



GERÇEK ve YARI GERÇEK ZAMANLI YÜZ TESPİT ETME

Atakan KAPLAN

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr.Öğr.Üy. Ahmet ÇINAR

NİSAN-2018

T.C.
FIRAT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GERÇEK ve YARI GERÇEK ZAMANLI YÜZ TESPİT ETME

YÜKSEK LİSANS TEZİ

Atakan KAPLAN

(101129112)

Tezin Enstitüye Verildiği Tarih : 12.04.2018
Tezin Savunulduğu Tarih : 25.04.2018

Tez Danışmanı : Dr. Öğr. Üy. Ahmet ÇINAR (F.Ü.) 

Diğer Jüri Üyeleri : Prof. Dr. Ali KARCI (İnönü Üni.) 

Doç. Dr. Taner TUNCER (F.Ü.) 

NİSAN-2018

ÖNSÖZ

Yüksek lisans çalışmalarım süresince değerli bilgi veengin tecrübeleriyle yardımlarını esirgemeyen danışman hocam Sayın Dr.Öğr.Üy. Ahmet ÇINAR’a sonsuz teşekkürlerimi, şükranlarımı ve saygılarımı sunarım.

Ayrıca bana maddi ve manevi her türlü desteği veren aileme ve eşime en içten teşekkürlerimi ve saygılarımı sunarım.

Atakan KAPLAN

ELAZIĞ-2018

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ	II
İÇİNDEKİLER.....	III
ÖZET.....	V
SUMMARY.....	VI
ŞEKİLLER LİSTESİ	VII
TABLolar LİSTESİ	VIII
SEMBOLLER LİSTESİ.	IX
KISALTMALAR.....	X
1. GİRİŞ.....	1
1.1. Yüz Algılama Sistemi	2
1.2. Yüz Özellikleri	4
1.3. Yüz Algılama Verimini Etkileyen Faktörler.....	5
1.4. Tezi İçeriği.....	6
1.5. Tezin Amacı	6
2. YÜZ ALGILAMA YÖNTEMLERİ	7
2.1. Bilgi Tabanlı Yöntemler.....	7
2.2. Öznitelik Tabanlı Yöntemler	8
2.3. Şablon Eşleştirme Yöntemler	9
2.4. Görüntü Tabanlı Yöntemler.....	9
2.4.1. Doğrusal Alt Uzay Metodları.	10
2.4.2. İstatiksel Yaklaşımlar.....	10
2.4.3. Yapay Sinir Ağları.....	10
2.4.4. AdaBoost.	10
2.5. Yüz Algılama Sistemlerinde Genel Yaklaşım	14
3. MATERYAL ve YÖNTEM	21
3.1. Giriş	21
3.2. Haar-Cascade Sınıflandırıcı	21
3.3. Haar-Cascade Sınıflandırıcı Eğitimi.....	26
4. UYGULAMA	37
4.1. Uygulama İçin Kullanılan Yazılım ve Kütüphaneler.....	37
4.1.1. AForge.NET	37

4.1.2. EmguCV	38
4.2 Geliştirilen Sistem.....	38
5. SONUÇ	44
KAYNAKLAR.....	45
ÖZGEÇMİŞ	50



ÖZET

Son yıllarda, görüntülerde insan yüzünün algılanması araştırmacılar için ilgi çekici hale gelmiştir. Artan bu ilginin arkasındaki sebeplerden biri daha hızlı, gelişmiş ve havaalanlarında, stadyumlarda, hastanelerde ve fabrikalarda güvenliğin sağlanması için daha emniyetli bir düzen ihtiyacıdır. Hızlı ve etkin bir araç olan bilgisayarlar ile yüz algılama ve yüzün yerinin belirlenmesini kullanan birçok uygulama, hayatın zaruri bir parçası olmuştur.

Bu çalışmada, görüntü işleme teknikleri ve yüz bulma için Haar-Cascades Sınıflandırıcısı kullanılarak bir görüntünün yüz içerip içermediğinin tespiti, resim üzerinde yüz yerlerini saptama işlemleri gerçekleştirilmiştir. Bu teknikler uygulanırken Intel açık kaynaklı bilgisayar görme kütüphanesi (open source computer vision library - OpenCv) aracından yararlanılmıştır.

Proje kapsamında bulunan yazılım ve arayüzü Windows işletim sistemi ve Visual Studio çerçevesi üzerine C# programlama dili kullanarak hazırlanmıştır.

Anahtar Kelimeler: Yüz tanıma, Haar-Cascade Sınıflandırıcı, OpenCV, Görüntü İşleme

SUMMARY

FACE DETECTION ON REAL AND SEMI-REAL TIME

Human face detection in images has gained much interest in recent years, for the researchers. One of the reasons behind this increased interest is the need for faster, advance and more reliable tool to provide security at airports, stadiums, hospitals and factories. With computers as a powerful and fast tool, many applications that use face detection and localization are becoming an essential part of our life.

In that study, it is carried out to confirm the image that includes a face or not by using technique of image processing and Haar-Cascades Classifier for face detection, to find face places on the picture. When performing this methods, utilized from intel open source computer vision library (OpenCv).

Project software and its interface has been prepared by using C# programming language which is based on Windows operating system and Visual Studio framework.

Keywords : Face Detection, Haar-Cascade Classifier, OpenCV, Image Processing

ŞEKİLLER LİSTESİ

	<u>Sayfa No</u>
Şekil 1. Yüze ait bazı özelliklerin çıkartılması ve karşılaştırılması ile yüz tespiti.....	2
Şekil 2. Genel yüz algılama sisteminin temel birimleri.....	3
Şekil 3. Yüz metrik dataları.....	4
Şekil 4. Adaboost sınıflama örneği	11
Şekil 5. AdaBoost sınıflandırıcısı.....	12
Şekil 6. Cascade sınıflandırıcı işlem süreci	22
Şekil 7. Resmin İntegral edilmesi.....	24
Şekil 8. Haar Özellikler ve Uygulamaları(Burun ve Yanak)	25
Şekil 9. Haar Özellikler ve Uygulamaları	25
Şekil 10. Klasörlerin oluşturulması	27
Şekil 11. Negatif ve Pozitif klasörlerin oluşturulması.....	27
Şekil 12. Koordinat bilgileri	28
Şekil 13. Negatif.txt dosyası.....	30
Şekil 14. Vektör dosyası oluşturma	31
Şekil 15. .bat dosyasının çalıştırılması	32
Şekil 16. Doğrulama	33
Şekil 17. Dosya Doğruluk Kontrolü	33
Şekil 18. Eğitim aşaması	34
Şekil 19. Eğitici .bat dosyasının çalıştırılması.....	36
Şekil 20. dll dosyalarının eklenmesi.....	38
Şekil 21. Program arayüzü.....	39
Şekil 22. Resim yükleme	40
Şekil 23. Yüzlerin bulunması	40
Şekil 24. Video Yükleme	41
Şekil 25. Video dosyasında yüz tespiti.....	42
Şekil 26. Video akımının yapısı	42
Şekil 27. Kameradan yüz tespit	43
Şekil 28. gri-seviye dönüşümü yapılmış resim.....	43

TABLÖLAR LİSTESİ

	<u>Sayfa No</u>
Tablo 1. Yüz Tespit yaklaşımlarının kıyaslanması	13
Tablo 2. Haar-Cascade Sınıflandırıcı’da kullanılan Haar öznitelikleri	22



SEMBOLLER LİSTESİ

$f_{m,i}$: 2-boyutlu integrallerin ağırlıklı toplamı

$\alpha_{m,i}$: i . öznitelik çıkarıcı

$\beta_{m,i}$: i . öznitelik çıkarıcı

$\theta_{m,i}$: m . sınıflandırıcı için karar eşiği



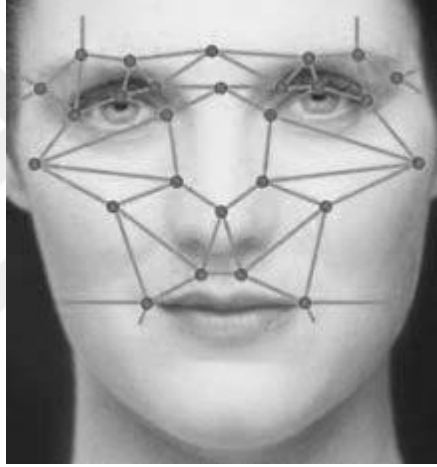
KISALTMALAR

RGB	: Red, Green, Blue
HSV	: Hue, Saturation ve Value
DAA	: Doğrusal Ayırma Analizi
FA	: Faktör Analizi
TBA	: Temel Bileşen Analizi
LBP	: Local Binary Pattern
SVM	: Support Vektor Machine
GST	: Generalized Symmetry Transform
FMSD	: Fuzzy Maximum Scatter Difference
MSD	: Maximum Scatter Difference
SAT	: Sum Area Table
KNN	: K-Nearest Neighbor
HOG	: Histogram of Oriented Gradient
PLBP	: Principal Local Binary Pattern
LBP	: Local Binary Pattern
DCT	: Discrete Cosine Transform
YSA	: Yapay Sinir Ağları
GMM	: Gizli Markov Modeli
NBS	: Naive Bayes Sınıflandırıcısı

1.GİRİŞ

İnsan tanıma ve kimliklendirme günümüzün gelişen toplum ve teknolojiyle önem kazanmış ve önemli bir bilgi sistemi haline gelmiştir. İnsan beyninin aşamalı öğrenme sistemi yapay sinir ağlarında da denenmiş, yüz tespitinde başarıyla kullanılmıştır [1]. Çalışmacılar özetledikleri makalelerde araştırmacıları uyarıyorlar: "Yüz tespit algoritmaları ve sistemleri tasarlayanlar psikofiziksel ve nörofizyolojik bulgulardan haberdar olmalı, ama sadece pratik anlamda işe yarayacak olanları modellerinde kullanmalıdırlar" [2]. Kişi tanıma; bir kişinin fiziksel karakteristikleri veya kişisel özellikleri kullanılarak otomatik olarak kimliğinin belirlenmesi veya doğrulanması işlemidir. Bu işlemin otomatik olması kullanıcının sisteme ya çok az müdahale etmesi ya da hiç müdahale etmemesi anlamına gelir, temel olarak o kişiye ait yüz, el şekli, parmak izi, retina, DNA ve ses ana başlıkları altında sayılan biyometrik özellikler kullanılarak yapılabilir. Yüz tespit sistemleriyle tutarlı olma iddiası taşıyan ilk yüz tespit modeli Turk ve Pentland'ın özyüz (eigenface) modelidir [3]. Diğer bir yaklaşım da elastik çizge (elastic graph) modelidir [4]. Bu modelde Gabor dalgacıkları ile yüzlerden yüz hatları çıkartılmak suretiyle tespit yapılır. Ramasubramanian ve Venkatesh yüz imgelerine DCT uygulamışlardır [5]. Özellikle biyometrik tabanlı kimliklendirme sivil toplumda veya yasal düzenleme bulunan alanlarda kullanılır. Ama yüz algılamada bu özellikleri ne şekilde dağıldığı, hangisinin altta, hangisinin üstte olduğu, yani özelliklerin yapılandırılması, özelliklerin kendisinden daha önemli bir bilgidir [6]. İnsanın görsel olarak karşısındakini tanımada kullandığı en yaygın yöntem, yüz tanıma, günümüzde en çok kabul gören biyometrik tabanlı tanımlamadır. Bu nedenle yüz tespit her dönemde güncelliğini korumuş ve üzerinde fazla sayıda çalışma yapılan bir konu olmuştur. Fakat öte yandan "Bu kadarcık yüz görerek yüzleri böylesine iyi tanımak mümkün değildir, yüz tanıma doğuştan var olan, kendine has bir sistem tarafından gerçekleştirilir" diyen bir grup var. Bu gruba göre yüz tespit Jerry Fodor'un zihnin modülleri olarak adlandırdığı izole fonksiyonlardan biri [7]. Geçmişten günümüze insan yüz tespit alanında ciddi gelişmeler olmuştur. Bu sebeptendir ki özellikle fiziksel güvenliğin kontrolün olduğu havaalanlarında, bankacılıkta, güvenlik kuvvetlerinin yapmış olduğu çalışmalarda kullanım alanı geniştir. Gelişen yöntem ve teknikler paralelinde diğer alanlarda da kullanım alanı yaygınlaşacaktır. Dahası ülkedeki birçok kamera vasıtasıyla otomatik yüz tespit işlemi film karelerinden çıkıp gerçek dünyada da yaygın kullanımı olacaktır.

Gelişim açısından bakılınca yüz tespit aşama aşama öğrenilen, her aşamada bir önceki aşamada kullanılan görüntülere yeni fonksiyonların eklendiği karmaşık bir sistem gerektiren bir problemdir [8]. Örneğin yaygın olan parmak izi tanımda deneğin hassas bir şekilde alınmış parmak izi ya da iris tanıma için deneğin yüksek çözünürlüklü bir kamerada yakın mesafeden göz resimleri gerekir. Ayrıca diğer biyometrik tanımla sistemlerinin kişilik haklarını ihlal etmesi (parmak izi vs.) ve bazı biyometrik izlerin alınması kanunla saklı tutulmuş gerekli mahkeme kararı ve yasal zorunluluklar durumunda kayıtları alınabilmektedir. Bu sayılan sebepler dolayısıyla yüz biyometrisi çalışmaları daha bir önem kazanır [9].



Şekil 1.1: Yüze ait bazı özelliklerin çıkartılması ve karşılaştırılması ile yüz tespiti

1.1 Yüz Algılama Sistemi

Bir yüz algılama sistemi, görüntü bilgisinden belirli özellik dizisini süzen, bu özellik dizisini diğer özellik dizileriyle karşılaştıran ve bu karşılaştırma sonucuna dayanarak karar verme kuralına göre yürüten bir yapıya sahiptir. Tipik bir yüz algılama sistemi şekil 1.2’de görülmektedir. Altı birimden oluşan sistem içeriğinde: veri edinim birimi, altölçeklendirme birimi, önışleme birimi, özellik çıkarım birimi, sınıflandırma birimi ve çözümleme birimi yer almaktadır. Bu birimlerin her biri aşağıda açıklanmıştır:

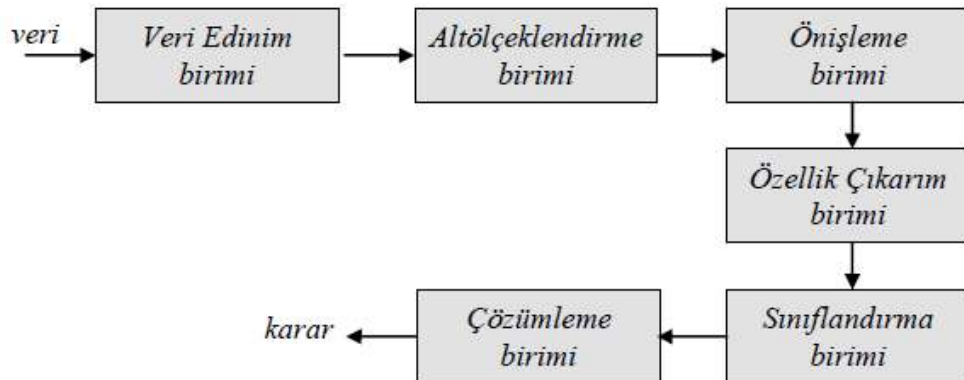
Veri Edinim Birimi: Bir görüntünün ham bilgisinin uygun bir kamera veya tarayıcı ile elde edilmesi gerekir. Veri edinim birimi insan makine arayüzünü tanımlar, bundan

dolayı, yüz algılama sisteminin performansı için çok önemlidir. Kullanılan kameranın teknik özellikleri de edinilen veri kalitesi açısından çok önemlidir.

Altölçeklendirme Birimi: Resim çokça yüz içerebilir. Bu yüzler farklı boyutlarda ve farklı uzaklıkta olabilir. Bu nedenle orijinal resmin alt örnekleme yapılmış bir sürümü oluşturulur ve değişik boyuttaki diğer resimleri algılamak için ayrı olarak işlem yapılır.

Önişleme Birimi: Sistem tarafından ele edilen verinin kalitesi istenilen düzeyde olmayabilir. Bunun için, sonraki işlemler için uygunluğunu tanımlamak için ilk olarak hesaplama yapılır. Buna ek olarak, elde edilen veri kalitesini artırmak için bir sinyal iyileştirici uygulanır. Bazı önişleme yöntemleri şunlardır:

- **Aydınlanma Normalleştirme:** Bu önişleme, görüntü tanıma tabanlı sistemler için spesifik bir işlemdir. Bu yöntemin amacı, gözlemlenen resmin farklı alanlarda yakalandığında aydınlanma etkilerini azaltmaktır.
- **Histogram Eşitleme:** Resmin her katmanında eşit sayıda piksel içermesi için yapılan işleme histogram denkleştirme adı verilir.
- **Ortanca Süzme:** Bu yöntem basit ve çok etkili bir şekilde gürültü giderme işlemidir. Normal olarak görüntüdeki gürültüyü azaltmak için kullanılır.
- **Yüksek-geçirgen Süzgeçleme:** Yüksek geçirgen süzgeçleme, görüntüdeki düşük frekanslı bileşenlerini yüksek frekans gradyanları etkilemeden ortadan kaldırır [10].

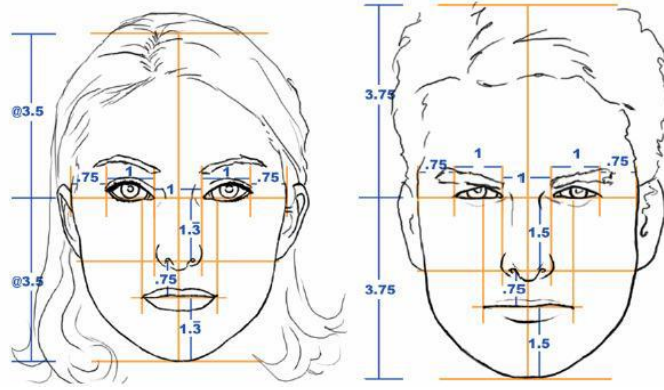


Şekil 1.2: Genel yüz algılama sisteminin temel birimleri

1.2 Yüz Özellikleri

Vücut oranlarında olduğu gibi insan yüzünde de tespit edilmiş belirli oranlar vardır. İnsan yüzünde yer alan bazı oranlar şöyledir [11]:

- Yüz uzunluğu / Yüzün eni,
- Kaşların ve dudak bütünleşim yeri arası / Burun uzunluğu,
- Yüzün uzunluğu / Kaşların ve çene ucu bütünleşim yeri arası,
- Ağız uzunluğu / Burun eni,
- Burun eni / Burun delikleri arası
- Göz bebekleri arası / Kaşlar arası



Şekil 1.3: Yüz metrik dataları

Yüz ile beraber insan kafasına ait bazı oranlar ise;

- Gözler kafa uzunluğunu tam ortasında bulunur.
- Burun, ağız ve çenenin eni neredeyse aynıdır.
- Bir gözün eni yüz eninin 0.4 katı kadardır.
- İki göz arasındaki mesafe bir gözün uzunluğu kadardır.

İnsan yüzündeki özellik noktalarının bulunması göz bebeğinin belirlenmesiyle gerçekleştirilir. Yüz hatlarının bulunmasında belirlenmiş oranları aşağıdaki gibidir(Şekil 1.3) [12];

- Kafa uzunluğu erkeklerde 7.5 kadınlarda birim 7 birim olarak hesaplanmıştır.
- Göz kısmı kafa uzunluğunun yaklaşık olarak yarısı uzunluğunda konumlanmıştır.
- Kafa eni erkeklerde ve kadınlarda 4.5 birim olarak hesaplanmıştır.

- Burnun ve gözlerin eni 1 birim ölçüsündedir.
- Çene ve dudak arası uzunluğu erkeklerde 1.5, kadınlarda ise 1.3 birim uzunluğundadır [13].

1.3 Yüz Algılama Verimini Etkileyen Faktörler

Yüze ait öznitelikler, yüz görüntüsünün görünümünü düzelteren faktörlere karşı değişmez özellikte olmalıdır. Bu farklılıklar aşağıdaki faktörler ile ifade edilebilir.

- *Duruş Farklılıkları:* Kameranın duruşu değişkenlik gösterdiği için, yüz görüntüsü aşağıdaki etkilere bağlı olarak değişebilir: a) izdüşümsel bozulma (yayılp genişlemeye ve yüzün bir bölümünün yanında küçük kalmaya sebep olur) ve b) yüzün bir bölümünün kapanması. Eğer yüzleri sadece bir bakış açısından görürsek, genellikle diğer açılardan algılanmaları zor olur.
- *Aydınlanma:* Poz değişikliği ile birlikte aydınlanma farklılıkları kaçınılmazdır. Gün içinde ortamın ışık yoğunluğu bina içi ve dışında değişiklik göstermektedir. Yüzün 3 boyutlu formuna bağlı olarak, doğrudan uygulanan aydınlatma kaynağı yüze ait belirli bazı özelliklerin vurgulanmasına veya belirginliğinin azalmasına sebep olan koyu gölgeleri oluşturabilir. Bu durum insanlar için problem teşkil etmezken, bilgisayar açısından büyük sorunlara sebep olabilir.
- *Yüz Gösterimi:* Yüz değişmez bir nesne değildir. Duygulanma sonucunda yüzde oluşan ifade ve konuşurken oluşan mimikler yüz görüntüsünde büyük değişikliklere sebep olabilmektedir. Yüz ifadesi, yüz özelliklerinin geometrik biçimini ve konumunu etkilemektedir. Bu etki, geometri tabanlı algoritmalar için daha büyük iken; bütünsel algoritmalarda daha azdır.
- *Kapanma (Occlusion):* Görüntü içinde yüzün bir bölümü diğer nesneler, güneş gözlüğü veya diğer gereçler tarafından kapanabilir. Bu kapanma kasıtlı olabileceği gibi kasıtlı olmayabilir.
- *Yapısal Bileşenler:* Sakal, bıyık, farklı büyüklük ve renkteki gözlükler gibi yüze ait özellikler yüz algılama performansını etkilemektedir.
- *Başın Yönelimi:* Yüz görüntüleri, fotoğraf makinesinin teknik özelliklerine bağlı olarak değişebilir.

1.4 Tezin İçeriği

Bu çalışmada gerçek ve yarı gerçek zamanlı yüz tespit sisteminin çeşitli teknikler kullanılarak yüzün metrik datalarının kullanılması çerçevesinde gerçekleştirilmiştir. Bu tezin ikinci bölümünde “Yüz Tespit” yöntemlerinin genel olarak neler olduğu anlatılmış ve bu bölümlere ait bilgiler verilmiş, geçmişten günümüze “Yüz Tespit” üzerine literatürde yapılmış olan çalışmalar derlenmiş ve bu çalışmanın amacından bahsedilmiştir. Üçüncü bölümde kullanılan metotlar hakkında bilgiler verilmiştir. Haar-Cascade Sınıflandırıcı hakkında genel bilgiler sunulmuştur. Sonraki bölümde ise sözü edilen yöntemler ve teknikler kullanılarak yüz tespitine yönelik uygulama yapılmıştır. Sonuç bölümünde daha etkili ve verimli yüz algılama için neler yapılabileceğinden bahsedilmiştir.

1.5 Tezin Amacı

Bu çalışmada resim, video ve web cam aracılığıyla yüz algılanması hedeflenmiştir. İnsanların herhangi bir zorluk çekmeden bir yüzü tanıyabildikleri ve bir yüzün makinelerce olabildiğince doğru ve hızlı bir şekilde algılanması ancak hızlı ve güvenilir algoritmalar kullanılarak mümkün olacağı inkâr edilemez bir gerçektir.

Bu amaç doğrultusunda, yapılan çalışmada, literatürde sık rastlanılan birçok algoritma kullanılarak, hızlı ve doğru yüz bulunması için kapsamlı bir inceleme gerçekleştirilmiştir. Çalışmanın temeli, farklı yüzlere ait yüz görüntülerine ilişkin özniteliklerin Haar-Cascade Sınıflandırıcı kullanılarak elde edilmesi üzerine kurulmuştur.

2. YÜZ ALGILAMA YÖNTEMLERİ

Yüz algılama ve izleme son yıllarda çok yaygın bir araştırma konusu olmuştur. Bir yüzün sınıflandırılabilmesi için öncelikle görüntüdeki “yüz” kısmının belirlenmesi gerekmektedir. Görüntüler genellikle kişilerin ön yüzlerinden çekilmiş şekildedir. Buradaki en önemli problem, görüntüdeki yüz bölgesine ait kısmın doğru bir şekilde çıkarılmasıdır. Çünkü yüz bölgesinin konumu ve büyüklüğü her bir görüntü için farklılık gösterebildiğinden, görüntülerden yüz kısmının çıkarılmasında aynı türden bir şablon kullanılması pek mümkün olmamaktadır [14]. Bu nedenle yüz kısımları, çeşitli yüz belirleme teknikleri [15,16] veya el ile yapılan çalışmalar sonucu çıkarılmaktadır. Belirgin ve kesin bir yöntem olmadığından resimden yüz bilgisini çıkartma işleme el ile yapılsa dahi kesin olarak yüzde yüz çıkartılması mümkün olmamaktadır. Sözelimi farklı boyutlarda iki resimdeki yüz resimleri çıkarttığımızda, çıkarılan yüzlerin boyutları da farklı olma ihtimali %99.9”dur ve boyut eşitlenirse dahi matematiksel olarak her bir yüz resmin matris değerleri boyut olarak farklı olacaktır [17]. Birçok yaklaşım ve algoritma kesin ve sağlam sonuçlar elde etmek için önerilmiştir. Genel olarak yüz algılama yöntemleri dört kategoriye ayrılmıştır.

- Bilgi tabanlı yöntemler
- Öznitelik tabanlı yöntemler
- Şablon eşleme yöntemleri
- Görüntü tabanlı yöntemler

2.1 Bilgi tabanlı yöntemler

Bu yöntem yüz bilgisinden elde edilen önceden tanımlanmış kurallara dayanarak geliştirilmiştir. Örneğin, bir görüntüdeki yüz genellikle birbirine simetrik şekilde duran gözler, bir burun ve bir ağız olarak görünür. Özellikler arasındaki ilgili mesafe ve konum aralarındaki ilişkiyi ifade edebilir. Yüz görüntüsünün tespitinde aday görüntü bu kodlanmış kurallara göre tespit edilir.

Yang ve Huang”ın araştırmasında bu yöntem kullanılarak yüz algılama için hiyerarşik bir bilgi tabanlı yöntem üzerinde çalışılmıştır [18]. Bu çalışmanın genişletilmiş bir versiyonu Kotropoulos ve Pitas [19], tarafından ön cephe görüntülerinde çoklu çözünürlük fikrinin kullanılmasıyla yapılmıştır.

2.2 Öznitelik tabanlı yöntemler

İnsan gözü, yüzleri farklı pozlar ve farklı aydınlanma koşullarında kolayca algılayabilir. Tüm bu farklı koşulların yanında değişmeyen nitelikler de olmalıdır. Bazı yöntemler, ilk olarak kenar algılayıcılar ile kaş, göz, burun, ağız ve saç çizgisini algılar ve daha sonra yüzü bularak teyit eder. Ten rengi de özellik olarak kullanılabilir. Çıkarım yapılmış özniteliklere bağlı olarak, aralarındaki ilişkiyi tanımlamak ve yüzün varlığını kanıtlamak için istatistiksel bir model oluşturulur. Öznitelik tabanlı algoritmalarındaki olası bir olumsuzluk ise görüntü özniteliklerinin aydınlanma, gürültü ve kapanmalara bağlı olarak bozulmaya uğramalarıdır. Özniteliklerin sınırları gölgelerin güçlü kenarlar oluşturmasına bağlı olarak belirsizleşir. Bu durum da algoritmaların yetersiz kalmasına sebep olur.

Öznitelik tabanlı sistemler üzerinde çalışan Sirohey [20], kenar haritalarını kullanmıştır. Leung [21], yüz algılama için olasılıklı bir yöntem kullanarak yüzü tanımlamak için burun deliği, göz ve dudak / burun ilişkisini kullanmıştır. Burl [22] ve Leung [23] şekillerin istatistiksel teorisinin kullanılması ile ilgili değişiklikler yapmışlardır. Ten renginin öznitelik olarak kullanıldığı birçok yöntem önerilmiştir.

RGB renk uzayı ten bölgelerini algılamak için kullanılmıştır [24]. Daha sonra, yüzün varlığının kanıtlanması için daha fazla işlem yapılmaktadır. Bazı çalışmalarda düzgelenmiş RGB renk uzayındaki değerlerin histogramı kullanılmaktadır [25]. Sobottka ve Pitas'ın [26], çalışmasında HSV renk uzayı ve yüz özellikleri, yüzün çıkarımı ve yerinin belirlenmesi için kullanılmıştır. HSV uzayında tene benzeyen alanların bulunması için renklerin ayrılması işlemi yapılmış daha sonra bu alanların varlığı içlerindeki yüz özelliklerinin bulunmasıyla kanıtlanmıştır.

Diğer bir renk uzayı çalışması Chia ve Ngan [27] tarafından yapılmıştır. YCrCb renk uzayı renkli görüntülerin yüz ile ilgili alanlarının yerlerinin belirlenmesi için kullanılmıştır. Öncephe yüzlerini algılamak için önerilen bir sistemde ise Simetri Tabanlı Değer Fonksiyonu Algılama (Symmetry based cost function detection) kullanılmıştır. Burada yüz özelliklerinin çıkarımı için YES renk uzayı da önerilen sistem ile birlikte kullanılmıştır [28].

2.3 Şablon Eşleme yöntemler

Şablon eşleme yöntemlerinde, standart bir ön cephe yüz şablonu önceden tanımlanır veya bir fonksiyon ile ifade edilir. Verilen bir giriş görüntüsünde, ilinti değerleri standart bir şablon ile yüz hatları, göz, burun ve ağız için birbirinden bağımsız olarak hesaplanır. İlinti değerlerine dayanarak yüz olup olmadığına karar verilir. Bu yaklaşımın uygulama kolaylığı sağlaması avantajının yanında yüz algılama için ölçek, poz ve şekil değişikliklerinden dolayı algılama başarısının düşüşü söz konusudur. Çoklu çözünürlük, çoklu ölçek, alt şablonlar ve biçim değiştirebilen şablonların kullanılmasının önerilmesiyle değişmeyen ölçek ve şekil elde edilmesi hedeflenmiştir. Tsukamoto [29], görüntünün her örneğinin blok ve özelliklere (parlaklık ve kenar) bölündüğü ve onların her blok için kestirim yapılması önerilmiştir.

2.4 Görüntü tabanlı yöntemler

Bu yöntemlerin büyük bir kısmında düşük boyutlardaki bir pencere belli niceliklerde küçültülmüş görüntü frameleri(çerçeve) üzerinde uygulanarak insan yüzleri aranır. Yüzleri bulmak için görüntüden alınan parçalar eğitilen veya insan eliyle hazırlanan modellerle karşılaştırılır [30].

Bu yaklaşımlar yüzün yer değiştirme ve pozuna karşı oldukça duyarlıdır. Bir görünüm tabanlı yüz tespit süreci aşağıdaki ana adımları içermektedir [31]:

1. Yüz tespit edilecek resimlere ön bir işlem yapılarak yüz bulmaya uygun şekle getirilmesi.
2. test ve eğitme görüntülerinin tek biçime getirilmesi.
3. Algoritmanın negatif ve bazen de pozitif giriş bilgileriyle eğitilmesi.
4. Yüzlerin tespiti için bir arama izleminin uygulanması.

Alınan ilk görüntülerdeki ön bir işlem temel olarak verilen giriş görüntülerinin renk yoğunluğu, edge(kenar), arka plan deseni, boyut, şekil, renk çeşitliliği ve karışıklık dağılımı gibi özelliklerinin tek bir standarda getirilmesini amaçlar. Bu adım özellikle farklı yüz doğrultuları, farklı parlaklık koşulları ve karışık background(arka plan) çalıştırılması amaçlanan yüz tespit uygulamaları için oldukça önemli ve kritiktir.

Görünüm tabanlı yüz bulma yöntemlerinde çoğunlukla şekil tanıma, makine öğrenmesi ve veri madenciliği konuları kapsamında geliştirilmiş iyi bilinen sınıflandırma algoritmaları kullanılır.

Bilinen sınıflandırma algoritmaları şunlardır:

1. TBA
2. DAA
3. FA
4. GMM
5. AdaBoost
6. İstatistiksel Dağılım Tabanlı Metotlar
7. SVM
8. YSA
9. Tümevarımsal Öğrenme
10. NBS

Bu algoritmaların gruplanmış hali aşağıdaki gibidir:

2.4.1. Doğrusal Alt Uzay Metotları

Gri seviyeli yüz görüntüleri çok boyutlu tüm resim alanları içinde bir alt uzayı oluştururlar. Aslında bütün sınıflandırıcılar ile bu alt uzay temsil edilmeye çalışılır. FA, DAA ve TBA insan suretlerinin oluşturduğu alt uzayı simgelemek için kullanılan yöntemlerdir.

2.4.2. İstatistiksel Yaklaşımlar

NBS ve SVM sınıflandırıcı örnektir. Özellikle SVM iki sınıflı problemlerin çözümünde başarılı olduktan sonra birden fazla sayıda yüz tespit algoritmasında kullanılmıştır.

2.4.3. Yapay Sinir Ağları

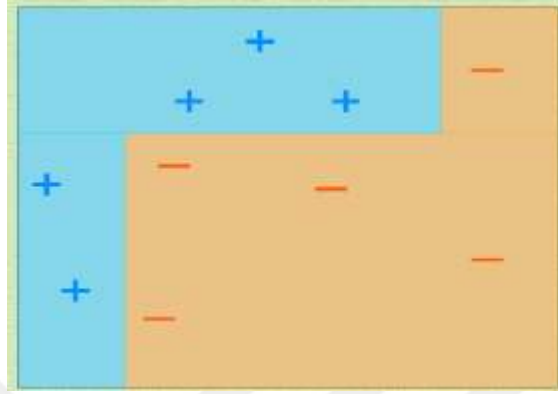
YSA, insan beyninin çalışma mekanizmasını taklit ederek beynin öğrenme, hatırlama genelleme yapma yolu ile yeni bilgiler türetebilme gibi temel işlevlerini gerçekleştirmek üzere geliştirilen mantıksal yazılımlardır. YSA biyolojik sinir ağlarını taklit eden sentetik yapılardır. YSA yavaş çalıştığından özellikle süre sınırlamasının olmadığı birçok uygulamada kullanılmış ve başarılı sonuçlar elde edilmiştir [32].

2.4.4. AdaBoost

AdaBoost algoritması kullanılarak bir çok başarılı yüz tespiti yapılmıştır. Bu algoritma sınıflandırma, cinsiyet ve yüz tespit gibi iki sınıflı problemlerde başarılı

olmuştur. Schapire ve Singer AdaBoost'u çoklu sınıf ve çoklu etiket sürümüne genişletmişlerdir.

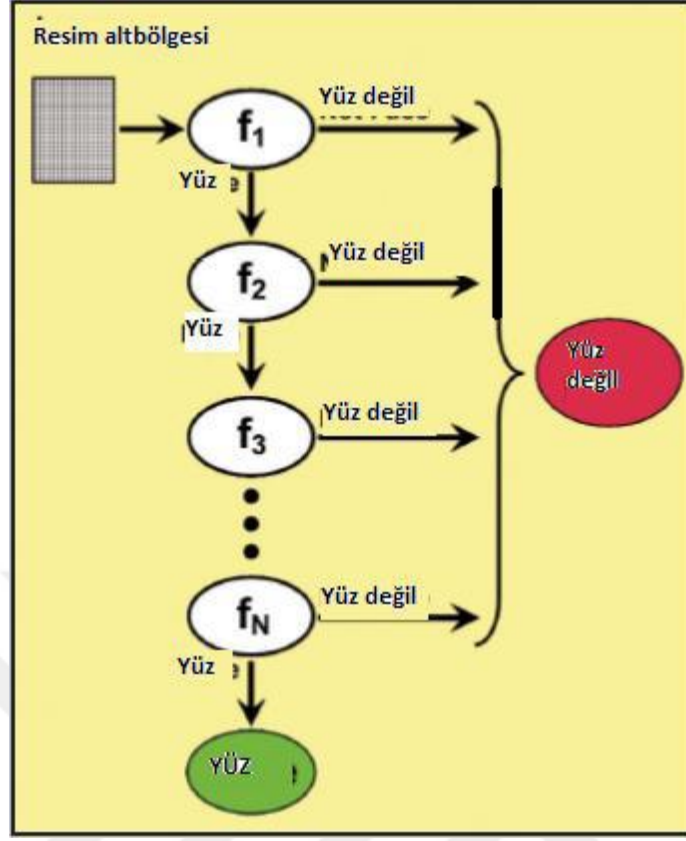
Şekil 2.1'de AdaBoost ile sınıflanmış iki bölümlü bir örnek gösterilmiştir.



Şekil 2.1: Adaboost sınıflama örneği

Algoritmanın amacı, eğitim örnekleri üzerinden hesaplanan bir D dağılımına bağlı olarak zayıf sınıflayıcı oluşturmaktır. D dağılımı algoritmanın eğitim setindeki her bir örneğe verdiği katsayıların kümesidir.

AdaBoost algoritması çalışmaya her bir eğitim örneği için eşit bir D dağılımıyla başlar. Her adımda sınıflama performansına bağlı olarak en iyi zayıf sınıflandırıcı bulunur ve ağırlıklar güncellenerek bir olasılık dağılım fonksiyonu elde edilir. Sonraki adımda bu işlemler tekrarlanır, belirli sayıda iterasyon sonucunda en güçlü zayıf sınıflandırıcı bir araya getirilerek güçlü bir sınıflandırıcı oluşturulur. Şekil 2.2'de AdaBoost sınıflandırıcısı görülmektedir.



Şekil 2.2 : AdaBoost sınıflandırıcısı

Yüz tespit etme işleminde, bir alanın yüz içermesi için tüm sınıflayıcılardan başarılı olması gerekmektedir.

Yüz tespit yaklaşımlarının artı ve eksileri Tablo 2.1 ile verilmektedir.

Tablo 2.1 : Yüz Tespit yaklaşımlarının kıyaslanması

Yaklaşım	Özellik Tabanlı	Görünüm Tabanlı	Şablon Eşleme	Bilgi Tabanlı
Özellikler	Bölgesel	Bütünsel	Şablon	Tanımlanmış
Karmaşıklık	Kolay	Zor, uzun eğitim süreci	Kolay	Kodlaması zordur
Görüntü Adedi	Tek resim olsa dahi yüksek başarı	Resim sayısı fazla olmak zorunda	Tek resim olsa dahi yüksek başarı	Görüntüye göre değişkenlik gösterir
Boyut	Düşük boyutlu	Çeşitli yöntemlerle boyut azaltılmalıdır	Fark etmez	Fark etmez
Bellek Kullanımı	Minimum	Maksimum	Kalıp boyutu kadar	Kurallar doğrultusunda
Çevresel Duyarlılık	Az	Çok	Orta	Kurallara bağlı
Başarım	Düşük başarı	Yüksek başarı	Orta	İyi
Avantajlar	Yüze ait poz doğrultuları ve aydınlanma koşullarından bağımsızlardır.	1. İspatlanmış başarılı sonuçlar üretirler. 2. Güçlü ve başarılı makine öğrenmesi algoritmalarını kullanırlar. 3. Farklı boyutlardaki ve doğrultulardaki yüz resimleri için de başarı ile çalışırlar. 4. Hızlı ve etkin çalışırlar.	Hesaplama açısından oldukça basit yöntemlerdir.	1. Karmaşık olmayan arka planlarda oldukça başarılı sonuçlar üreten yöntemlerdir 2. Kodlanmış kurallar baz alınarak, bir test resmi içindeki yüze ait özellikleri çıkarmak ve sonra aday yüzleri onaylamak basit yöntemlerdir. 3. Yüze ait özellikleri ve bu özelliklere ait ilişkileri tanımlayacak kurallar bulmak çok kolaydır.
Dezavantajlar	1. Karmaşık arka planlar üzerindeki yüzlerin bulunması zordur. 2. Aydınlanma ve resimlerdeki diğer benzeri gürültüler, yüzlerdeki özelliklerin bulunmasını zorlaştırır.	1. Sınıflandırma algoritmalarını eğitmek için birçok pozitif ve negatif örnek gerektirirler. 2. . Genellikle resim üstünde tarama işlemi gerektirirler.	1. Farklı poz doğrultuları için şablon yüzlerin bulunması ve doğru olarak uygulamak zor. 2. Şablonlar eğer yüze yakın bölgelerden itibaren test resimleri üzerinde taranmazlarsa yüzün bulunması çok maliyetli olabilir.	1. Farklı pozlardaki resimler için bu yöntemlerin uygulanması oldukça güçtür. 2. İnsan bilgisini kurallara dönüştürmek her zaman kolay olmayabilir. Çok detaylı kurallar kullanıldığında yüzleri bulmak mümkün olmayabilir, daha az detaylı kurallar kullanıldığında da hatalı pozitif sonuçlar bulunabilir.

2.5 Yüz Algılama Sistemlerinde Genel Yaklaşım

Otomatik yüz tespit 1960 yıllarında geliştirilen yeni bir kavramdır. İlk yarı-otomatik yüz tespit sistemi Woody Bledsoe, Helen Chan Kurt ve Charles Bisson tarafından oluşturulmuştur. Bu sistemde, ortak bir referans noktasının mesafelerinin ve oranları hesaplanmadan önce fotoğrafların özelliklerini (örneğin, gözler, kulaklar, burun, ağız vb.) elle belirlenip ve bu belli noktaları belirledikten sonra, sistem geometrik mesafeleri ve eğimleri hesaplayarak sonuç verilmiştir.

Goldstein vd. [33], yazarlar tarafından saç rengi ve dudak kalınlığı gibi 21 özel öznel belirteçleri kullanılarak otomatik yüz tespit sistemi gerçekleştirilmiştir. Yüz tespit ile ilgili bu ilk iki çözümde, ölçümlerin ve yerlerin elle hesaplanması gerekir ve bu yüzden çok emek ve zamanı gerektiren bir sistemdir.

Sirovich ve Kirby [34], yazarlar tarafından yüz tespit problemi için standart lineer cebir tekniği olan TBA tekniğini uygulanmıştır. Yüzü doğru ve uygun bir şekilde hızlandırmak ve normalleştirmek için, yüzden daha az değer gerektirmesinden dolayı bu teknik bir dönüm noktası olarak kabul edilmektedir.

Türk ve Pentland [35], yazarlar tarafından özyüzler (eigenfaces) teknikleri kullanılırken görüntülerdeki yüzleri algılamak için, kalan hatasının kullanılmasını önerilmiştir. Bu çalışma ile güvenilir ve gerçek zamanlı otomatik yüz tespit sisteminin geliştirilmesi mümkün olmuştur. Önerilmiş yaklaşımın uygulanması biraz çevresel faktörler tarafından kısıtlanmış olmasına rağmen, yine de otomatik yüz tespit teknolojilerinin geliştirilmesi için önemli altyapı sağlamıştır. Ocak 2001 tarihinde Super Bowl tarafından bu teknolojiyi kullanılarak gerçekleştirilen bir deneme uygulaması kamuoyunun dikkatini çekmiştir. Yazar gözetim görüntüleri çekmiş ve şahısların dijital fotoğraflarının tutulduğu veritabanı ile karşılaştırmıştır. Bu gösteri, halkın sosyal ve gizlilik meraklarını dikkate alarak ulusal ihtiyaçları desteklemek için bu teknolojinin nasıl kullanılabileceği hakkında çok ihtiyaç duyulan analizi başlatmıştı. Bugün, yüz tespit teknolojisi pasaport sahtekarlarını ve kanunu ihlal edenleri tespit etmek, kayıp çocukları tespit etmek ve kimlik dolandırıcılığını azaltmak için kullanılmaktadır.

Vetter ve Poggio [36], makalelerinde tek bir örnek görüntüden doğrusal nesne sınıfları ve görüntü sentezi geliştirmiştir. Yazarlar tarafından önerilen lineer sınıf kavramı genişleterek bir teknik geliştirilmiştir. Bu çalışma doğrusal nesne sınıfları için lineer dönüşümlerin tam olarak iki boyutlu prototip görünümünden oluşan temel setten öğrenilebilirliğini göstermiştir. Yazarlar söz konusu yaklaşımın yapay nesneler üzerinde

uygulayarak göstermişler ve sonra ön kanıt olarak önerilen tekniğin etkili olarak tek iki boyutlu görünümünden elde edilen yüksek çözünürlüklü yüz görüntülerini "döndürebildiğini" göstermişler. İki boyutlu modellere dayalı önerilmiş yaklaşım, herhangi bir derinlik bilgisi gerektirmemektedir, bu yüzden iki boyutlu görüntülerden üç boyutlu modeller üretmek bazen zordur. Farklı yönlerdeki nesneleri temsil eden görüntüler arasında hiçbir uyuşmaya ihtiyaç duyulmadığı için tam otomatik algoritmalar uyuşma bulma adımı olarak uygulanabilmektedir. Nesne tanıma görevleri için onların yaklaşımı çeşitli imalara sahiptir. Sadece tek bir görüntü verildiğinde önerilen teknikler yapay örnek görüntüleri oluşturulmaktadır. Öte yandan, şekil ve desenin örnek şekiller ve desenler gibi dekompozisyonu sonucunda elde edilen katsayılar her hangi üç boyutlu afin dönüşümü ile geçerli olmayan nesnenin temsilini vermektedir.

Sherrah vd. [37], makalelerinde değişen ana pozları kullanarak benzerlik uzayında yüz dağılımı üzerinde başarılı bir araştırma yapmışlardır. Önerilmiş sistemde görüntülerin prototipe benzerliğine dayanan statiksel öğrenme tekniği ile elde edilen görüntü kullanılmaktadır. TBA ve Gabor Filtresi kullanan dönüşüm gerçekleştirilmektedir. Bu dönüşümle benzerlikteki farklılığı dikkate almayarak pozdaki farkı gösterilmektedir. TBA ile boyut indirgeme, poz varyasyonunu doğru ifade etmenin yanı sıra kimlik konusunda değişmez bulunmaktadır. Önerilmiş araştırmada poz varyasyonunun geliştirilebileceğini gösterilip ve benzerlik uzay çerçevesi dahilinde masrafsız algoritmaların kullanımı ile kimlik benzerliklerinin gizli tutulabileceğini gösterilmektedir.

Bae ve Kim [38], tarafından yüz uzayı ve yüz hatlarından çıkarılan hibrid bilgileri kullanarak gerçek zamanlı yüz algılama ve tanıma sistemi önerilmektedir. Bu çalışmada, yüz tespitte yüksek performans elde etmek için ışık efektlerinin azaltması amaçlanmaktadır; çünkü yüz tespit ışıktan kaynaklanan nedenlerden dolayı ortaya çıkan değişikliklerle baş edememektedir. Önerilmiş mimaride iki ana teknik kullanılmaktadır: 1) yüz algılama için özyüzler uzay yöntemi ve 2) yüz tespit için yapay sinir ağların yöntemi. Gauss fonksiyonuna dayalı normalleştirilmiş cilt rengi haritası bir yüz aday alanını çıkarmak için uygulanmaktadır. Daha sonra aday alandaki yüz özellik bilgileri yüz bölgesini algılamak için kullanılmaktadır.

Ahonen vd. [39], tarafından yüz tespit uygulamaları için Yerel İkili Desenlerin (Local Binary Pattern – LBP) ile yüz açıklamaları tanıtılmıştır. Önerilen sistemde, yerel ikili desen ile doku özelliklerine göre verimli ve yeni yüz görüntü gösterimi sunulmaktadır. Önerilen yaklaşımda, ilk olarak yerel ikili desenleri dağılımları çıkarılan ve birleştirilmiş

gelişmiş bir özellik vektörü ile ilk yüz görüntüsünün alanlara bölünmektedir. İkinci olarak, yerel ikili desen tabanlı yüz tanıma oluşturulmaktadır. Bunların sonucunda, yerel ikili desenlerinin en iyi performansa gösteren doku tanımlayıcılarından biri olarak görünmektedir. Yerel ikili desenlerinin kullanılmasının avantajlarından ve bunun kullanıldığı çeşitli uygulamalardan bahsedilmektedir.

Yang vd. [40], makalelerinde renkli görüntü serisi ile bir yüz algılama ve tanıma sistemi üzerinde çalışmışlardır. Önerilmiş sistem iki alt sistemden oluşmaktadır.

Birincisi, insan yüz algılama alt sistemi, ikincisi ise insan yüz tanıma alt sistemidir. İnsan yüz algılama alt sisteminin iki modülden oluştuğundan bahsedilmektedir: yüz bulgu ve yüz doğrulama. İnsanın yüzü bulma modülünde, insan yüzünü ten rengi analizi ve hareket analizi ile tespit edilmişlerdir. İnsan yüzü doğrulama modülünde ise, destek vektör makinesi (Support Vektör Machine - SVM) tarafından tespit, insan yüzlerini doğrulamak için göz ve ağız yerleri ile insan yüzlerinin lokalize dönüşümünü ele almışlardır. Kısaca aşağıdaki teknikleri kullanılmışlardır. (a) özellik seçimi ve çıkarma için TBA. (b) SVM ve geometrik özellikleri yüz bölgesi doğrulama için kullanılmışlardır. (c) Cilt Chroma ile yüz tanıma için kullanılan hareket analizi yapmışlardır. (d) Cilt analizi ve hareket analizi. (d) Genelleştirilmiş Simetri Dönüşümü (Generalized Symmetry Transform - GST) yöntemi yüz, göz lokalize gibi noktaları ve özellik lokalizasyon simetri tanımlamak için kullanılmışlardır.

Singh vd. [41], makalelerinde tek galeri görüntüleri ile yüz tespit üzerine çalışmalar yapmıştır. Yazarlar iki boyutlu günlük polar Gabor dönüşümü kullanarak yüz doku fazını özelliklerini ayıklamak için dinamik sinir ağı mimarisini kullanmışlardır.

Wang vd. [42], bulanık maksimum dağılım ayırma analizi kullanılarak yüz tespit tasarlamışlardır. Bulanık maksimum dağılım farkı (Fuzzy Maximum Scatter Difference - FMSD) ayırt edici kriterdir, yeni bir alt uzay olarak örneklerden özellikleri ayıklamak için önerilmiştir. Bu algoritmanın, klasik Fisher diskriminant analizinde görünen küçük örneklem büyüklüğü sorununu ve sınıflar içi dağılım matrisinin teklik sorununu önleyerek geleneksel maksimum dağılım fark yöntemi esasına dayandığından bahsedilmektedir. Ayrıca, aykırı örnekler dağıtım bilgileri sınıflandırması için önemli olan, ilgili dağılım matrisinin yeniden tanımlanmasını yapmışlardır. Yaptıkları deneysel sonuçlar daha iyi tespit oranı elde etmişlerdir ve bu sonucun doğruluğunu da bulanık küme teorisi ile MSD (Maximum Scatter Difference)'yi kullanarak göstermişlerdir. Bu yöntemler doğrusal olma temeline dayandığı için veriler üzerinden doğrusal olmayan özellikleri

ayıklamak için başarısız olmuşlardır. Bu sorunu çözmek için de, bir çekirdek tabanlı alternatif FMSD yöntemini kullanmışlardır. Böylece, çok daha yüksek bir sınıflandırma doğruluğu sonucu elde etmişlerdir.

Kaminski vd. [43], makalelerinde tek bir yüz görüntüsü kullanılarak yüzün konumunu ve bakış algılamasını hesaplayan sistem önermişlerdir. Yazarlar önerilen sistemle aşağıdaki dört amacı gerçekleştirmektedirler. İlk amaca uygun olarak her bir görüntüden çıkartılan iki veya üç özellik noktaları gerektiren iki yeni algoritma önerilmektedir. İkinci amaç kapsamında ise tek bir görüntüden başın konumunu ve bakışın algılanmasını hesaplayan bir sistem önerilmektedir. Dördüncü amaç kapsamında tipik bir PC üzerinde gerçek zamanlı olarak çalışan sağlam bir algoritma önerilmektedir. Baş konumunun yaklaşımı ve de göz bakışın algılanması geometrik modeline dayanan bir yaklaşımdır. Şöyle kolay ve müessir bir yaklaşım önerilmektedir. İmplementasyonda basit ayarlar kullanılmaktadır. Bu basit ayarlarda kullanıcı kalibrasyonu hiç gerektirmeyen, tek sabit odaklı kamera kullanılmaktadır. Bu algoritma kapsamında baş konumu hesaplamak için göz algılaması hesaplanmaktadır. Ayrıca aşırı durumda ki eğim kamera eksenini ve kişinin yüzü arasında o kadar büyük ki gereken yüz özellikler kamera tarafından algılanmamaktadır. Bu aşırı durum için, tek bir gözün algılanmasının hesaplanması ikinci algoritma önerilmektedir. Önerilen sistem monoküler olduğu için birden fazla kameralar ile ilgili sorunlar kaçınılmaktadır. Kamera parametreleri, zaman içinde sabit muhafaza edilmektedir. Bu sonuç çıkarılan antropometrik özellikleri yüz ve göz modeli kullanılarak elde edilmiştir.

Echardt vd. [44], makalelerinde pratik yüz özelliğinin algılanması önermişlerdir. Yüz özelliğinin lokalizasyonu ve algılanması doğru bir yüz tespit ve algılama sisteminde anahtar bir öneme sahiptir ve bu sadece normal durumda değil, farklı koşullar altında tespit, mesela; poz varyasyonu, aydınlatma varyasyonu, yüz ifadesinin değişimi, ölçek değişimi vb; yüz özelliğinin lokalizasyonu ve algılanması çok önem taşımaktadır. Pratik uygulamalarda daha önemli olarak daha geniş koşullar altında mesela, cinsiyet, yaş, etnik, köken, poz ve aydınlatma farklılıkları dahil olmak üzere, güvenilir bir dedektör gerektirmektedir. Bu dedektörlerin gelişimi için, sağlamlık ve hassasiyet arasında denge kurmak ve daha önemlisi seçmek bir sorun olarak kabul edilmektedir. Sağlam dedektörler zayıf bir lokalizasyon sağlayıp, ama yerel yapısındaki küçük değişikliklere duyarlı dedektörler, çok sayıda yanlış alarm üretmektedir. Şimdi bu iki özellik arasında en doğrusunu seçmek için, yazarların önerilen sistemin bağlamında, bağımlı çıkarıma göre bir

yaklaşım sunulmuştur. İlk olarak, sağlam dedektörleri hedef özellikleri olduğu bağlamları tespit etmek için kullanılmıştı ve sonra hassas dedektörleri tespit bağlamda verilen özelliklerinin yerleştirilmesine eğitilmiştir. Önerilmiş yaklaşım en doğru lokalizasyon performans ile gerçek zamanlı olarak çalışmıştır.

Kim vd. [45], makalelerinde gerçek zamanlı yüz algılamak için dikdörtgen özelliği dayalı bir sınıflandırıcı önermişlerdir. Önerilmiş sistemde, hesaplama ve algılama performansı, hem de verimliliği tatmin etmek için güçlü bir algılama algoritma gerçekleştirilmiştir. Önerilen algoritmada özellik çıkarımı, sınıflandırıcı çalışma ve gerçek zamanlı yüz etki alanı algılama için üç aşama önerilmektedir. Özellik çıkarma için, özelliği oluşturmada önerilen beş dikdörtgen özellikleri ile belirlenen bir özellik düzenlenip ve özetlenme alan tabloları (Sum Area Table - SAT)'ı kullanarak verimli olarak özellik değerlerini hesaplanmaktadır. Sınıflandırıcı çalışması için, sınıflandırıcı öğrenme, Adaboost adlı bir algoritmayı kullanarak hiyerarşik bir sınıflandırıcı yaratılmaktadır. Ayrıca, bir sonraki düzeyde tekrar tekrar önemli yüzey desenleri uygulayarak mükemmel algılama performansı başlanmaktadır. Gerçek zamanlı yüz etki alanı algılama için oluşturulan dikdörtgen özelliği dayalı sınıflandırıcı ile hızlı ve verimli bir şekilde yüzün etkisi bulunmaktadır. Ayrıca, tespit oranı girdiği görüntüsü olarak TBA ve KNN (K-Nearest Neighbor) algoritmaları nokta tekniği mevcut noktasından daha sınıf içi bir sınıfı kullanarak bir yüz etki alanı, tespit alanı kullanılarak geliştirilmiştir.

Shi vd. [46], makalelerinde gerçek zamanlı yüz algılama ve tanıma sistemleri üzerinde çalışmışlardır. Yazarların amacı, yüksek tespit oranı ile görüntü veya videonun hızlı işlenmesidir ve bunun için önerilen sistemde üç anahtar yapı kullanmıştır. Birincisi, lineer Haar özellik kullanılmıştır. İkincisi, özelliklerin çıkarımı için TBA kullanılmıştır. Üçüncüsü ise, Haar özelliği kullanılırken yüksek hata algılama oranı dikkate alınarak çalışmalar yapılmıştır. Çünkü insan vücuduna ait olmayan görüntüler algılanmamalıdır. Bundan dolayı, bir video veya resimlerden insanın vücudunu algılamak için Odaklı Gradient Histogram (Histogram of Oriented Gradient - HOG) algoritması kullanılmaktadır. HOG algoritmasındaki temel düşünce, bir görüntünün şekil yoğunluğu geçişleri veya kenar tarifi dağılımı ile tarif edilmesidir. Bir nesne saptanırken, HOG özellikleri tanımlayıcı olarak kullanılmaktadır. Bu tanımlayıcılar görüntüyü hücrelere bölüp ve her hücre için gradyan yönleri veya kenar dağılımları doğrultusunda histogramlar oluşturulmaktadır. Bloklar içerisinde bulunan değerler normalize edilmektedir. Bununla birlikte, saptama

esnasında aydınlatma veya gölgeleme değişikliklerinin sonucu etkilemediği görülmektedir. Sonuç olarak, HOG insan algılamada kullanılan verimli bir algoritmadır.

Li vd. [47], yazarlar tarafından sağlam bir yüz tespit tekniği için medyan dayalı maksimum dağılım farkı (Maximum Scatter Difference - MSD) önerilmektedir. Önerilmiş sistemde, mevcut medyan dayalı (MSD)'nin zayıflığı bir ideal olmayan durumda mesela; poz ifadesi, aydınlatma ve yaş aşmak için çalışılmaktadır. Önerilen MSD modelinde, sınıfın ifadelerde ortalama vektörü dağılım matrisi sınıf içinde ve arasında sınıf saçılım matrisi sınıf örnek ortalama tahmin edilmektedir. İdeal olmayan şartlar altında, örnek sette aykırı olmaktadır, böylece sınıf örnek ortalama sınıfın doğru tahmin sağlamak için yeterli olmadığı verilen örneklerin birkaçı kullanarak ortalama yetmemektedir. Sonuç olarak, geleneksel MSD modelinin tanınma oranı azalmaktadır. Bu sorun (küçük örnek boyutu) ele almak için, medyan tabanlı MSD yerine (ortalama) dayalı maksimum dağılım kullanılmaktadır. Deneylerde birçok özellik çıkartma yöntemleri ile iki popüler veri setleri (CAS - PEAL veritabanı ve FERET veritabanı) üzerinde yapılan deney sonuçlarını göre klasik MSD'den gelişme gösterilmiştir.

Mishra vd. [48], yazarlar tarafından özyüzleri ve TBA'yı kullanarak iki boyutlu görüntülerinin yüz tespiti için bir yaklaşım önerilmiştir. Yüz görüntüleri normalde dik olacaktır, ama iki boyutlu özelliklerinin görüşleri, küçük bir set ile tanımlanabilmektedir. Önerilmiş yaklaşımda yüz görüntüleri bilinen yüz görüntüleri arasındaki varyasyon kodlamak için bir özellik alanı veya yüz alanı tahmin edilmektedir. Sonra öngörülen özelliği boşluk veya yüz alanı 'özyüzler' olarak tanımlanabilip ve yüz görüntü kümesinin öz vektörleri ile oluşturulabilmektedir. Daha sonra, bu sürecin bir sonucu denetimsiz bir şekilde, yeni bir yüz tespiti için kullanılmaktadır. Önerilmiş yöntemde dikkat karşısında, sadece çıkarma değil, aynı zamanda basit ve teknik forma görüntüyü getirmek için, sadece etkin matematiksel hesaplamalar alınmaktadır. Ayrıca gerçek zamanlı yüz tespit sistemleri ile veri toplama donanım ve yazılım arayüzü kullanarak uygulanmaktadır.

Pujol, ve Garcia [49], yazarlar tarafından veri madenciliği araçları kullanarak yüz tespit için asıl yerel ikili desenler (Principal Local Binary Pattern - PLBP) sunulmuştur. Önerilmiş teknik, geleneksel yerel ikili desen (Local Binary Pattern-LBP)'nin bir gelişmiş versiyonudur. PLBP'nin amacı LBP operatörü kullanarak yüz tespit işlemi gerçekleştirmek için gerekli özellik vektörlerinin boyutları azaltmaktır. Orijinal LBP operatörü, iki temel sorunu olduğu için, operatör ile bir yüz tanımlamak için gerekli hesaplama karmaşıklığı ve bazı ağırlıkları kesin bir şekilde seçilmemiştir. Önerilmiş çalışmada, bir yüz görüntüsü

karakterize, gerekli PLBP elde etmek ve daha sonra durumlarda veri madenciliđi araçları kullanarak, düzgün bölge ağırlıkları atamak için bir yöntem önerilmektedir. Sonuçta, bir yüz bilgileri içeren histogramlar boyutunda önemli ölçüde, % 94 bir tanıma oranı elde edilmiştir [50].



3.MATERYAL ve YÖNTEM

3.1 Giriş

Bu bölümde tezde önerilen yöntem, kullanılan sınıflandırıcının ne olduğu, nasıl eğitildiği ile ilgili bilgi verilmiştir. Haar-Cascade Sınıflandırıcı algoritması tarafından çıkartılan özniteliklerinden bahsedilmiştir.

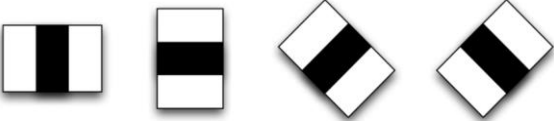
3.2 Haar-Cascade Sınıflandırıcı

Haar özellikleri, nesne belirlemede kullanılan sayısal görüntü özellikleridir. Bu özellikler ilk olarak real-time yüz tespiti için kullanılmıştır [51].

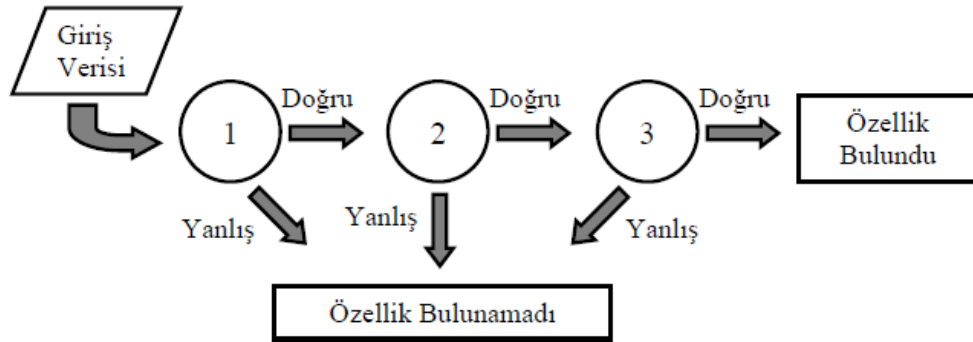
Haar özellikleri kullanarak nesne saptama yaklaşımı Haar dalgacık dönüşümünden adapte olunarak Paul Viola ve Michael Jones tarafından geliştirmiştir [52]. Temelde herhangi bir objenin alt parçacıklarının farklı renk dağılımına ve yoğunluğuna sahip olduğu bilgisi kullanılarak obje farklı alt parçalara ayrılır. Bu ayrılan parçalara ait özellikler farklı özellik setleri ile ifade edilerek objenin tamamı tanımlanır. Örneğin, insan yüzünde yanak gözler bölgesinden daha açıktır. Yanak ve göz bölgesinin her birine birbirine komşu iki dikdörtgen koyarak, dikdörtgenlerin altındaki bölgelerin piksellerin yoğunlukları farkı yüz belirlenmesinde kullanılır. Yüz üzerinde buna benzer başka özellikler de kullanılarak insan yüzünü diğer objelerden ayıracak özellik grubu inşa edilir. Elde edilen özellik grubu farklı eğitim setleri kullanılarak bir sınıflayıcı ile son nesne tanımlayıcıyı elde etmek üzere eğitilir. Nihai olarak eğitilen obje tanımlayıcı aynı tip objeleri, örneğin yaya ya da insan yüzü bulunması için kullanılır [53].

Veri filtreleme ve tanımlama gibi birçok uygulamaya sahip olan görüntü transformasyonu dalgacık dönüşümleriyle canlılık kazanmıştır. Birçok nesne dedektörü uygulamasında kullanılan Haar-Cascade Sınıflandırıcı bunlardan birisidir [54]. Sınıflandırıcı yüze ait kısımlardan alınmış, aynı boyuttaki resimlerle(pozitif resimler) ve yüz içermeyen diğer kısımlardan alınmış aynı boyuttaki (örn: 20x20) resimler (negatif resimler) kullanılarak eğitilir. Eğitilmiş sınıflandırıcıya yüz içeren bir resim parametre olarak verildiğinde yüzün bulunduğu alan için sınıflandırıcı “1” çıkışını diğer bölgeler için “0” çıkışını üretmektedir. Kullanılacak Haar benzeri öznitelik vektörleri Tablo 3.1’de verilmiştir.

Tablo 3.1: Haar-Cascade Sınıflandırıcı 'da kullanılan Haar öz nitelikleri

Öznitelik Çeşidi	Özellik Resimleri
Kenar(edge) öz nitelikleri	
Satır(line) öz nitelikleri	
Merkez-çevre(center-surround) öz nitelikleri	

Parametre olarak gelen resimdeki yüze ait alanların bulunması işlemi tüm resim taranarak gerçekleştirilmektedir. Sınıflandırıcı farklı boyutlardaki yüz özelliklerini bulacak şekilde planlanabilmektedir. Bunun avantajı ise görüntünün boyutlarını değiştirmeden işlem yaparak daha etkin bir yöntem olarak karşımıza çıkmaktadır. Bu nedenle parametre olarak gelen resim içerisindeki değişik ölçeklerdeki yüz özelliklerini bulabilmek için sınıflandırıcı aynı arama sürecini farklı ölçek parametreleri için tekrar eder.

**Şekil 3.1:** Cascade sınıflandırıcı işlem süreci

“Cascade” kelimesi, meydana gelen sınıflandırıcının basit yapılı birden fazla sınıflandırıcının bir araya gelerek kompleks yapılı bir yapı meydana getirdiğini göstermek amacıyla adlandırılmıştır.(şekil 3.1). Basit yapılı bu sınıflandırıcılar, herhangi bir sınıflayıcı katman geri çevirmediği sürece tüm katmanlar geçilene kadar parametre olarak verilen resme art arda uygulanır. Oluşturulan sistem içerisindeki her sınıflandırıcı katmanı

öznitelik uzayında farklı bir alanı taramaktadır. Burada kullanılan basit sınıflandırıcılar en az iki yapraklı karar ağaçlarıdır. Haar öznitelikleri bu sınıflandırıcılara giriş olarak verilir. Art arda zayıf sınıflandırıcıları eğitmek için FloatBoost ve AdaBoost yöntemleri kullanılabilir [55].

Haar sınıflandırıcı yardımıyla öncelikle yüz alanı belirlenmiştir. Akabinde tespit edilen yüz alanı içerisinde tekrar Haar-Cascade sınıflandırıcı çalıştırılmıştır. Haar-Cascade sınıflandırıcı bu şekilde peş peşe kullanılması yanlış tespitlerin önüne geçerek sistemin farklı ortam şartlarında çıktılarını doğru sonuç vermesini sağlamıştır[56].

Tezde real-time akan video görüntüsü veya resim üzerinde insan yüzü bölgelerinin yerinin belirlenmesinde boyut/yönelim değişimlerine karşı güçlü bir yöntem olan ve Haar özniteliklerini kullanan Haar-Cascade Sınıflandırıcı kullanılmıştır. Algoritmanın verimliliğini arttırmak için farklı boyutta resim ve resmin her alanına yüzlerce Haar özelliklerinin varlık ve yokluğunu tanımlamak amacıyla ilk olarak integral imaj üretilmektedir. Genel olarak “integral” kelimesinin anlamı, küçük birimleri birbiriyle toplamaktır. Bu durumda küçük birimler piksel değerlerimiz olmaktadır. Her bir piksel için integral değeri üst ve sol piksel toplamıdır. Parametre olarak verilen görüntüde sol üstten başlayarak sağa ve aşağıya tarama yapılarak her bir piksel birkaç tamsayı işlemi ile toplanmaktadır. İntegral imaj $ii(x,y)$ ile gösterilmektedir. Burada (x,y) imajın konumlandığı noktayı ifade eder. İntegral imaj denklem 1’de verilmiştir [57].

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y'), \quad (3.1)$$

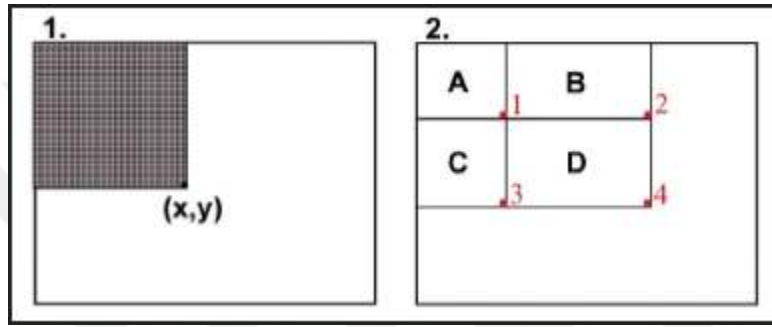
Denklemdaki $i(x,y)$, girdi olarak verilen imajdır. İntegral imaj, (x,y) koordinatının üstündeki ve sol tarafına doğru olan piksel değerlerinin toplamını içermektedir.

Haar-Cascade Sınıflandırıcı algoritmasının karar verme mekanizması denklem 2 ve 3’te verilmiştir.

$$C_m = \begin{cases} 1, & \sum_{i=0}^{l_m} F_{m,i} > \theta_{m,i} \\ 0, & otherwise \end{cases} \quad (3.2)$$

$$F_{m,i} = \begin{cases} \alpha_{m,i}, & \text{if } f_{m,i} > t_{m,i} \\ \beta_{m,i}, & \text{otherwise} \end{cases} \quad (3.3)$$

Burada $f_{m,i}$; 2-boyutlu integrallerin ağırlıklı toplamıdır ve aynı zamanda i . öznitelik çıkarıcı için karar eşiğidir. $\alpha_{m,i}$ ve $\beta_{m,i}$; i . öznitelik çıkarıcı ile bağıntılı sabit değerlerdir. $\theta_{m,i}$; m . sınıflandırıcı için karar eşiğidir. Şekil 3.2.'de (x,y) koordinatı üstünde ve solundaki koordinatların toplamından oluşan integral görüntü verilmektedir.

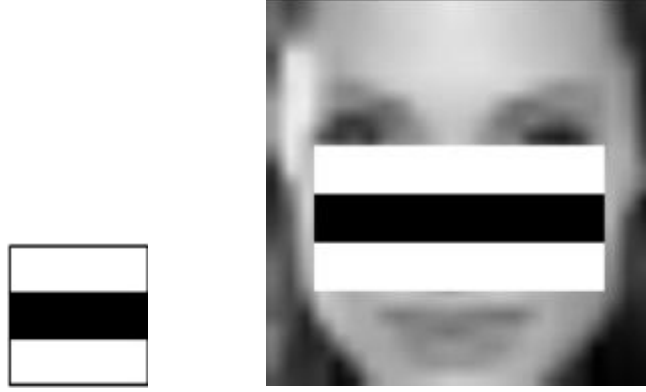


Şekil 3.2: Resmin İntegral edilmesi

Yüz tespiti işleminde pozitif resim içerisindeki nesneyi tespit etmek amacıyla tablo 3.1’de görüldüğü gibi 20x20 piksel boyutlarındaki alt-pencereler tüm resim boyunca kaydırılarak nesne taraması gerçekleştirilir. Bu pencereler, haar-benzeri öznitelikler olarak bilinmektedir

Feature denilen bu pencerelere zayıf sınıflandırıcılar denilmektedir; çünkü tek başına doğru bir sınıflandırıcı olamazlar. Bir nesnede bu zayıf sınıflandırıcılardan birçoğu olacaktır ve bu zayıf sınıflandırıcıların toplandığı noktada büyük doğruluk oranıyla aranan nesne var demektir. Sınıflandırıcı, en temel mantığıyla bu şekilde çalışmaktadır.

Pencereler aşağıdaki gibi örnek pozitif resimler üzerinde taranır.



Şekil 3.3 : Haar Özellikler ve Uygulamaları(Burun ve Yanak)

Yukarıdaki pencere için yanakların parlaklık oranının burun bölgesindeki parlaklık oranından daha düşük olması ile burun kısmı seçilebilir.



Şekil 3.4: Haar Özellikler ve Uygulamaları

Aynı zamanda göz bölgesinin beyaz ile gösterilen alt bölgeden daha karanlık olması da bu özelliklerden bir tanesidir(Şekil 3.4). Haar-Cascade Sınıflandırıcısında, buna benzer birçok özellikler içinde nesnenin bulunduğu resimler üzerinden geçirilerek değerler oluşturulur. Örneğin yüz taramasında ağız, burun, alın, saç gibi bölgelerde birçok karanlık aydınlık özellikleri oluşturulacaktır. Bunların her birinden hedef değerler oluşturulmaktadır ve bu işlem pencere büyüklükleri değiştirilerek diğer aşamalarda tekrar edilmektedir.

Bu pencereler(zayıf sınıflandırıcılar) her resim boyutu için düşünüldüğünde yüz binlerce çekirdek oluşacaktır. Negatif resimler üzerinde tarama yapılarak içinde nesne

bulunmadığı için kullanılmayacak olan pencerelerin büyük çoğunluğu elenecektir. Pozitif resimlerde nesne seçilerek nesnedeki kullanılacak pencereler belirlenecektir. Bunun için eğitim sırasında pozitif resimlerde nesnenin hassas seçimine dikkat edilmelidir. Pozitif ve negatif resim örneklerinin çok olması istenilen nesnede daha iyi sonuçlar almak için önemlidir.

Bu işlemlerin hem eğitimde hem de nesnelerin bulunmasında bilgisayarı çok yoracağı ve işlemlerin uzun süreceği düşünülebilir. Reel time görüntü işlemede hız çok önemlidir. Haar-Cascade Sınıflandırıcısında öncelikle resimlerin integrallerini aldığı için bilgisayardan büyük bir işlem gücü kaldırılacaktır. Ayrıca nesnelerin bulunması aşamasında da her bir büyüklükteki çerçeve tarafından tekrar tekrar taranmak yerine sadece önceki aşamalarla eşleşme olan kısımlar taranarak bir işlem yükü de oradan azaltılmaktadır. Bu açılarından ve yaptığı iş açısından oldukça hızlı oldukları söylenebilir de uygulama aşamasında reel time çalışıldığında alınan görüntüyü belirli oranda yavaşlattığı görülmektedir. Bu hız, sınıflandırıcının eğitime şekli, örnek sayısı gibi durumlara göre değişmektedir. İstenilen nesnelerin bulunabilmesi için sınıflandırıcının referans alacağı minhitrate ve maxfalsealarmrate değerleri vardır. Sınıflandırıcı eğitimde her seferinde belirli eğitim algoritması ile bu değerlere ulaşmaya çalışır. minhitrate minimum isabet oranını ve maxfalsealarmrate ise hangi hata oranındaki nesnelerin gösterileceğini ifade eder.

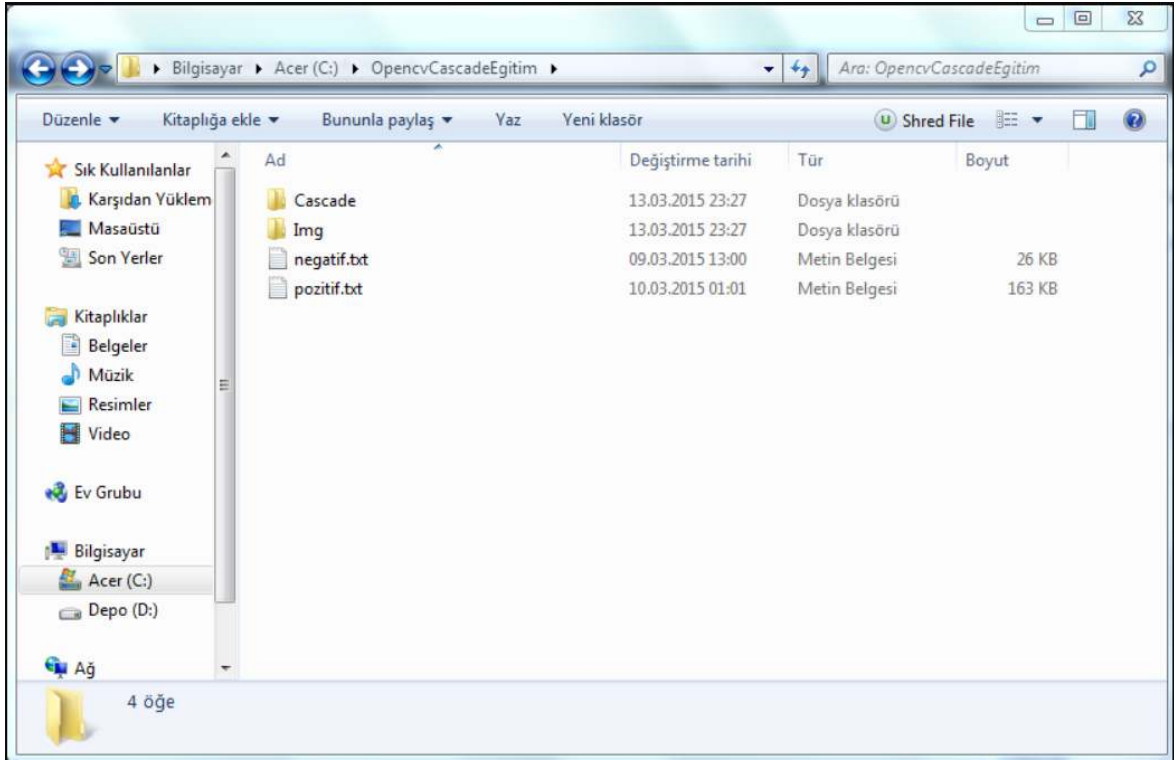
Eğitim yapıldıktan sonra .xml uzantılı bir dosya oluşturulacak ve bu dosya ile opencv kütüphanesi kullanılarak istenilen nesne bulunabilecektir.

3.3 Haar-Cascade Sınıflandırıcı Eğitimi

Haar-Cascade Sınıflandırıcı yöntemi kullanarak yüz, araç, el gibi istediğiniz türden farklı nesneleri sınıflandırıcıya tanıtırak farklı ortamlardaki resimlerde taratarak buldurabiliriz. Bu sınıflandırıcının eğitimi için pozitif ve negatif resimlere ihtiyacımız olduğuna değinilmişti. Sınıflandırıcımızı ne kadar çok örnek ile eğitirsek o kadar iyi sonuç elde edebiliriz. İyi bir sınıflandırıcı oluşturabilmek için sınıflandırıcının mantığıyla beraber kullanılan parametrelerin neler olduğunu anlamamız gerekmektedir. Pozitif resimler, içerisinde bulmak istediğiniz nesnelerin bulunduğu örneklerdir. Bu örnekler farklı parlaklıkta ve farklı açılarla olursa daha efektif bir sınıflandırıcı oluşturulabilir. İyi bir

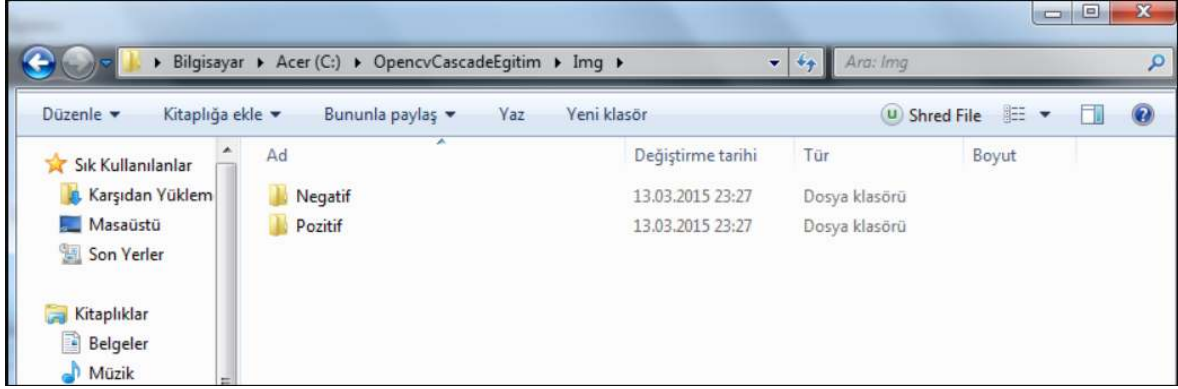
sınıflandırıcı için binlerce örnek ile eğitilmesi gerekmektedir; ancak daha az örnekle de sınıflandırıcı oluşturulabilir. Aşağıda adım adım sınıflandırıcının nasıl eğitildiği gösterilmiştir;

1-Öncelikle C:/ dizininde bir klasör oluşturulur ve içerisine “img” ve “Cascade” isimli klasörler ile “negatif.txt” ve “pozitif.txt” isimli text dosyaları oluşturulur.



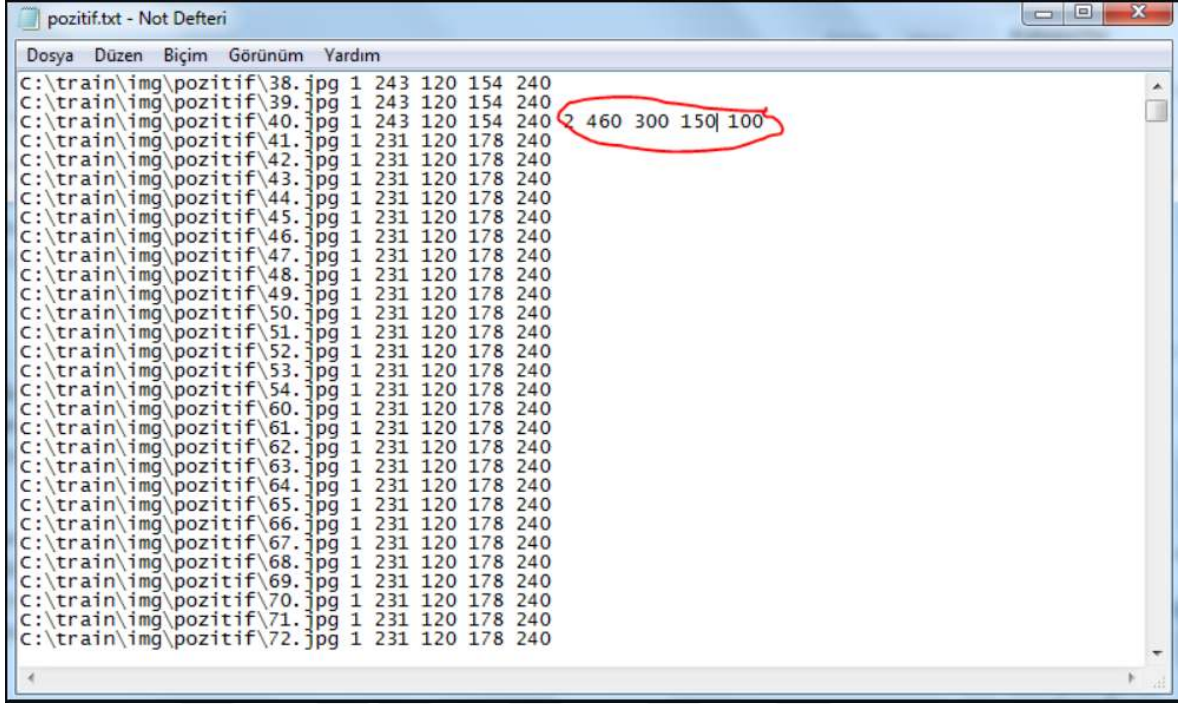
Şekil 3.5: Klasörlerin oluşturulması

Img klasörünün içine girerek “negatif” ve “pozitif” isimli klasörler oluşturulur.



Şekil 3.6: Negatif ve Pozitif klasörlerin oluşturulması

2- İçerisinde seçmek istediğiniz nesnenin bulunduğu örnek resimler “Pozitif” isimli klasörün içerisine kopyalanır. Resimler jpg, png,tiff,jpeg veya bmp uzantılarında olabilir. Ancak bütün uzantıların desteklenmediğini göz ardı edilmemelidir. Yüklediğimiz resimlerin yakın boyutlarda olduğuna dikkat edilmelidir. Bu aşamadan sonra ana dizinde “pozitif.txt” isimli dosyanın içerisine pozitif örnek resimlerinizin isimlerini dosya yoluyla beraber yapıştırılır. Örneğin şekil 3.6’da gösterilen “Pozitif” klasörünüzdeki ornek.jpg isimli resim dosyası için txt dosyasına C:\OpencvCascadeEgitim\Img\Pozitif\ornek.jpg bağlantısını kopyalanır. Fotoğraf editörleriyle ya da başka bir şekilde fotoğraftan seçilmesini istediğimiz nesne dikdörtgen içerisine alınır ve sol üst köşenin x ve y koordinatları ile genişlik ve yüksekliğini başında “1” sayısı ile beraber birer boşluk bırakarak kopyaladığımız bağlantının yanına yapıştırılır. Örneğin resmimizdeki nesnenin sol üst köşesi x=250 ve y=350 koordinatlarında olan ve genişliği 100, yüksekliği 200 piksel olan bir örnekte “pozitif.txt” dosyasına C:\OpencvCascadeEgitim\Img\Pozitif\ornek.jpg 250 350 100 200 yazılır. Eğer resmimizde birden fazla seçmek istediğimiz nesne varsa başına “2” yazarak onun da koordinat değerlerini yazmamız gerekmektedir.



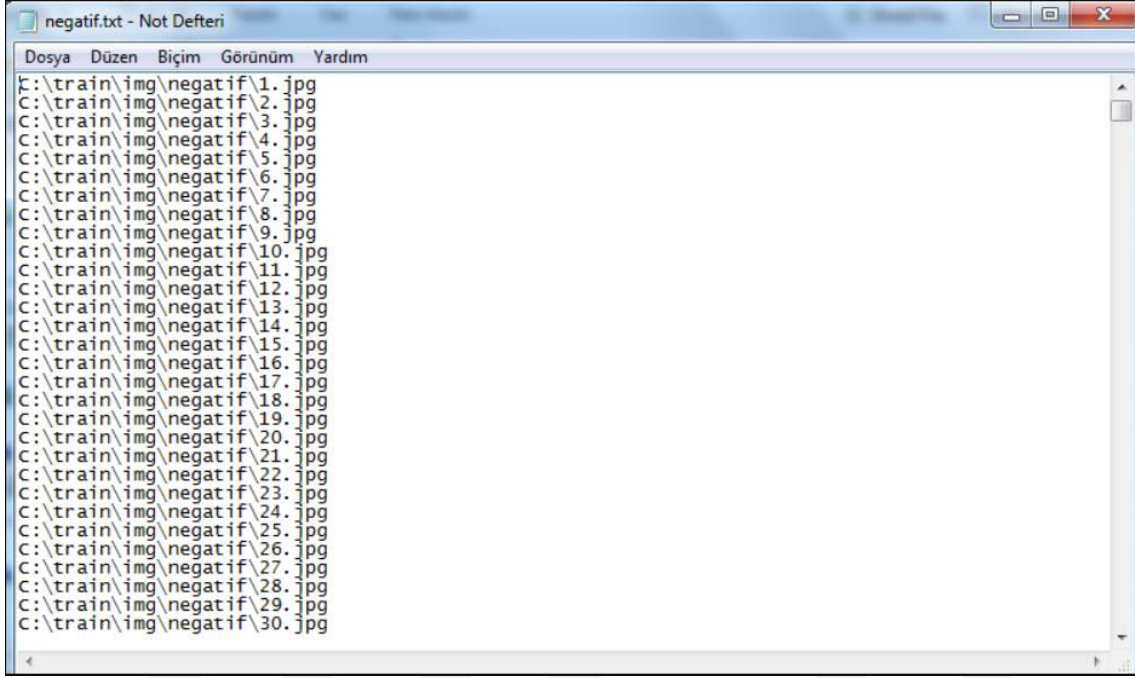
Şekil 3.7: Koordinat bilgileri

Şekil 3.7’de text dosyasında C:\train\img\pozitif\ klasöründeki resimler listelenmiş ve yanlarında içinde bulunan nesnelerin bulunduğu koordinatlar yazılmıştır. Kırmızı ile işaretli “2” ile başlayan kısım resimde ikinci bir nesnenin bulunduğunu ve koordinatlarını göstermektedir. Diğer nesnelerde aynı şekilde “2” sayısı artırılarak eklenebilir.

Resimlerimiz, OpenCV kütüphanesinde bulunan bir fonksiyon tarafından yan tarafında belirttiğiniz değerlere göre kırılacak ve onlar üzerinden işlem yapılacaktır. İçinde nesnenin bulunduğu bütün resimlerimizi yukarıda belirtildiği şekilde txt dosyasına alt alta eklemeliyiz; ancak resmin isim ve yolunda boşluk bulunmamasına dikkat edilmesi gerekmektedir. Eğer resimleri küçültmek için program kullanırsak otomatik olarak boşluk karakteri koyabilir. Resim adlarının aralarında boşluk olmamalıdır. Bu işlem hem yorucu hem de uzun süreçli biri işlem olduğu için buna yönelik pozitif resim oluşturucular türemiştir.

3-İçerisinde bulmak istediğimiz nesnenin olmadığı ve nesneyi arayacağınız ortamlara yakın ortamların resimlerini de “Negatif” isimli klasöre kopyalanmalıdır. Pozitif resimler için geçerli olan resim uzantıları negatif resimler için de geçerlidir ve örneklerin aynı boyutta olmalarına dikkat edilmelidir. Daha sonra “Negatif” klasöründe bulunan

resimlerimizin isimleriyle beraber yollarını yukarıda bahsedildiği gibi “negatif.txt” dosyasına alt alta yazılmalıdır. Ancak bu dosyada “pozitif.txt” de olduğu gibi kırpma parametreleri olmayacaktır, sadece resimlerin kendisi bulunacaktır. Örnek “negatif.txt” dosyası şekil 3.8’de ki gibi olacaktır:

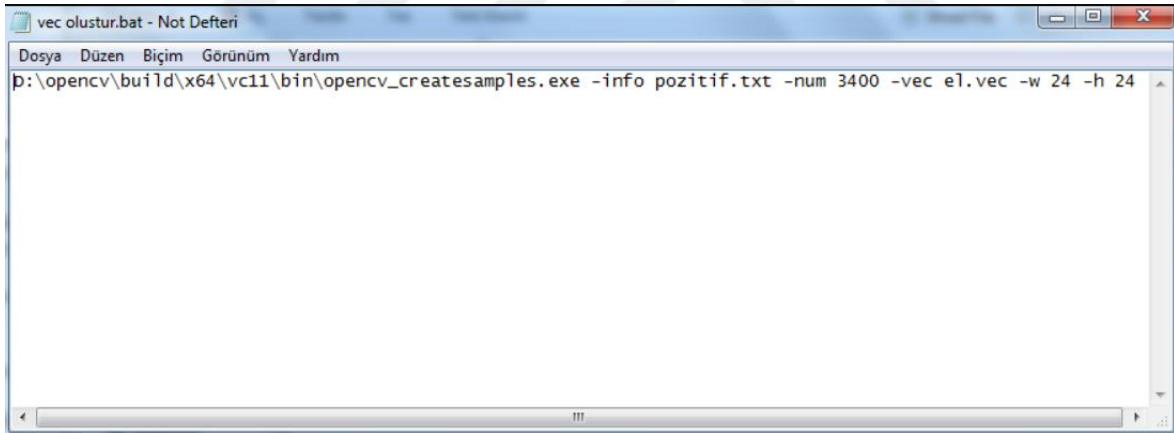


Şekil 3.8: Negatif.txt dosyası

Sınıflandırıcının efektif olarak nesneyi bulabilmesi için binlerce pozitif resme ihtiyacımız olduğunu aşikârdır. 2000 ile 4000 civarındaki pozitif resim sayısı iyi bir sonuç verebilir. Negatif resimlerin de 800 ile 1000 gibi bir adette olması iyi olabilecektir çünkü sınıflandırıcı pozitif resimleri negatif resimler üzerinde işlemektedir; ancak bu kadar fazla negatif resim zorunlu değildir. Daha az sayıyla da sınıflandırıcı oluşturulabilir.

4- Pozitif ve Negatif klasörlerimizi örneklerle doldurduktan ve txt dosyalarımıza bu örneklerimizin isim ve yollarını kaydettikten sonra, pozitif resimlerimizi Haar-Cascade Sınıflandırıcısında kullanabilmek için gri skalaya çevirmemiz ve nesnenin bulunduğu kısmı kırparak bir vektör dosyası oluşturmamız gerekmektedir. Bunun için opencv’yi kurduğunuz dizinde, D:\opencv\build\x64\vc11\bin dosya yolunda bulunan opencv_createsamples.exe dosyası işimizi kolaylaştıracaktır. 32 bit bilgisayarlar için x64 yerine x86’yı ve visual studio’nun farklı versiyonları kullanılıyorsa vc11 yerine bilgisayarda kurulu olan versiyonlarını seçmeleri gerekmektedir. Bu .exe dosyası

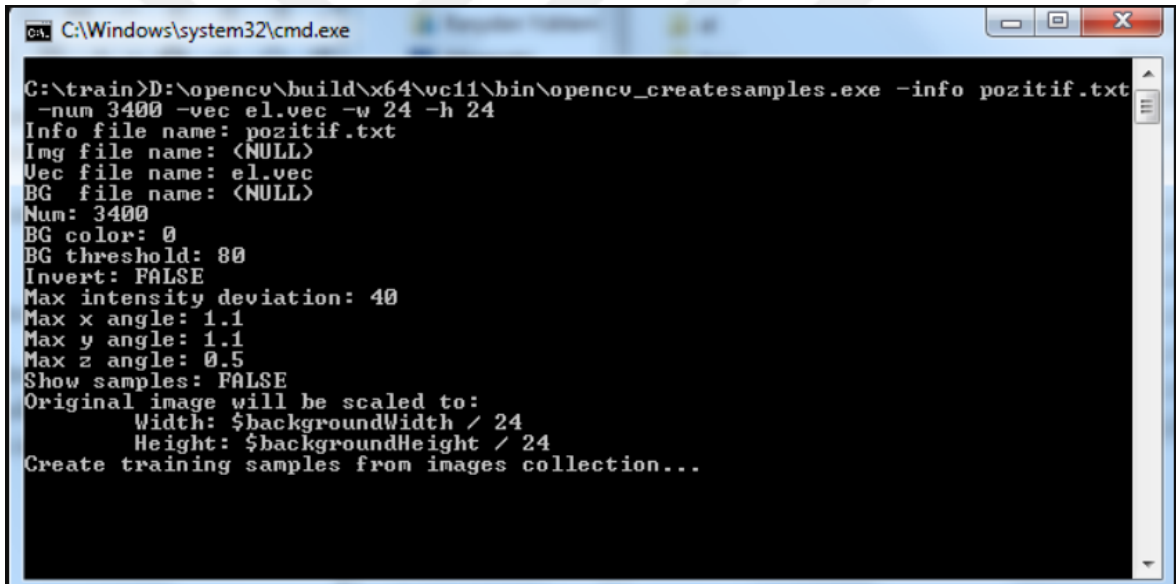
parametre olarak pozitif resimlerin isim ve adres yolunun bulunduğu txt dosyasını, pozitif resim sayısını, oluşturulacak olan vektör dosyasının ismini ve örneklerin boyutlandırılacağı oran değerlerini alır. Bu parametre değerlerini `opencv_createsamples.exe` dosyasına gönderebilmek için ya bilgisayarımızda bulunan command window çalışılır ya da .bat uzantılı bir toplu işlem dosyası oluşturulur ve command window'a yazılacak komutlar kaydedilerek dosya çalıştırılır. Sınıflandırıcı eğitimini tekrar tekrar kullanabileceğiniz düşünülürse .bat uzantılı bir dosya oluşturmak daha mantıklı olacaktır. “pozitif.txt” ve “negatif.txt” dosyalarının bulunduğu ana dizine yeni bir text dosyası oluşturulur ve .txt yazan kısmını .bat olarak değiştirilir. Oluşturduğumuz dosyaya sağ tıklayarak “düzenle” kısmına girilir. Text editörü ile açılan kısma `opencv`'yi kurduğumuz dizinde seçtiğimiz `opencv_createsamples.exe` isimli dosyamızı dosya yoluyla beraber kopyalayarak parametreleri girilir. Vektör dosyası oluştururken `D:\opencv\build\x64\vc11\bin\opencv_createsamples.exe -info pozitif.txt -num 3400 -vec el.vec -w 24 -h 24` şeklinde parametreler kullanılarak oluşturulur. Bu ifade şekil 3.9'da ki gibi .bat dosyasına yapııştırılarak kaydedilir. Parametrelerin karşılıkları şu şekildedir:



Şekil 3.9 : Vektör dosyası oluşturma

- `D:\opencv\build\x64\vc11\bin\opencv_createsamples.exe` kısmı `opencv` dizininde bulunan `opencv_createsamples.exe` dosyasının yoludur. command window bu komutu gördüğünde belirtilen yolda bulunan .exe dosyasını çalıştırır.
- `-info pozitif.txt` kısmı, `opencv_createsamples.exe` dosyasının ihtiyacı olan resimleri nerede bulacağı ve ne şekilde kıracağı bilgisini tutan, oluşturduğumuz txt dosyasını parametre olarak almaya yarar.

- -num 3400 kısmına “pozitif.txt” dosyamıza kaç tane örnek resim kaydettiyseniz o sayıyı girdiğimiz kısımdır. Vektör dosyasının kaç tane resimden oluştuğu bilgisi tutulur.
- -vec el.vec kısmına ise oluşturmayı istediğimiz vektör dosyasının ismini belirler. Vektör dosyasının .vec uzantısı ile biteceğine dikkat edilmelidir.
- -w 24 -h 24 bu kısım ise belirtilen koordinatlarda kırpılmış olan resmi genişliği 24, yüksekliği 24 piksel olacak şekilde yeniden boyutlandırır. Bu sınıflandırıcının çalışması için kolaylık sağlar. Binlerce pikselle çalışmaktansa 24*24 ten 576 piksel üzerinde çalışmak daha hızlı ve kolay sonuç verecektir. Bu boyutu seçmek istediğiniz nesneye göre değiştirebiliriz. -w genişliğini, -h ise yüksekliğini belirtir.



```

C:\Windows\system32\cmd.exe
C:\train>D:\opencv\build\x64\vc11\bin\opencv_createsamples.exe -info pozitif.txt
-num 3400 -vec el.vec -w 24 -h 24
Info file name: pozitif.txt
Img file name: <NULL>
Vec file name: el.vec
BG file name: <NULL>
Num: 3400
BG color: 0
BG threshold: 80
Invert: FALSE
Max intensity deviation: 40
Max x angle: 1.1
Max y angle: 1.1
Max z angle: 0.5
Show samples: FALSE
Original image will be scaled to:
    Width: $backgroundWidth / 24
    Height: $backgroundHeight / 24
Create training samples from images collection...

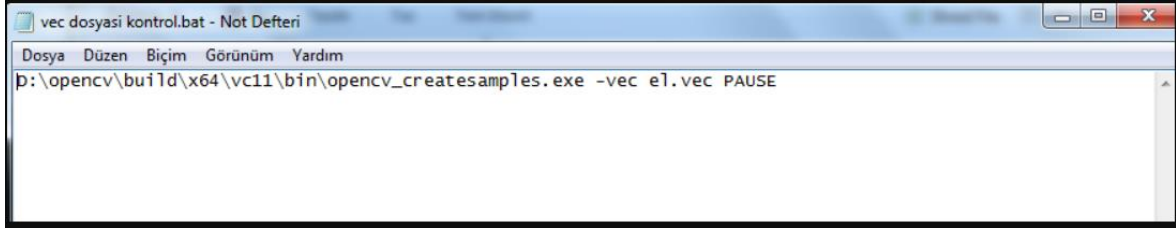
```

Şekil 3.10: .bat dosyasının çalıştırılması

Oluşturduğumuz .bat isimli dosyaya çift tıkladığınızda şekil 3.10’da ki gibi bir pencere görülür ve işlem bittiğinde kendi kendine kapanması gerekmektedir. Bu işlem .vec uzantılı vektör dosyası oluşturur.

5-Vektör dosyasını oluşturduktan sonra, bu dosyanın doğruluğunu kontrol etmek için yine opencv_createsamples.exe dosyasını kullanılır. Diğerinden farklı olarak vektör

dosyasını direkt parametre olarak verilir. Bu şekilde pozitif resimlerimizi görebiliriz. Aynı ana dizinde bir .bat dosyası daha oluşturulur ve şekil 3.11’de ki gibi düzenlenir:



Şekil 3.11: Doğrulama

Parametreleri kendinize göre güncelleyip dosyayı çalıştırdıktan sonra şekil 3.12’de ki gibi bir ekran görülür:

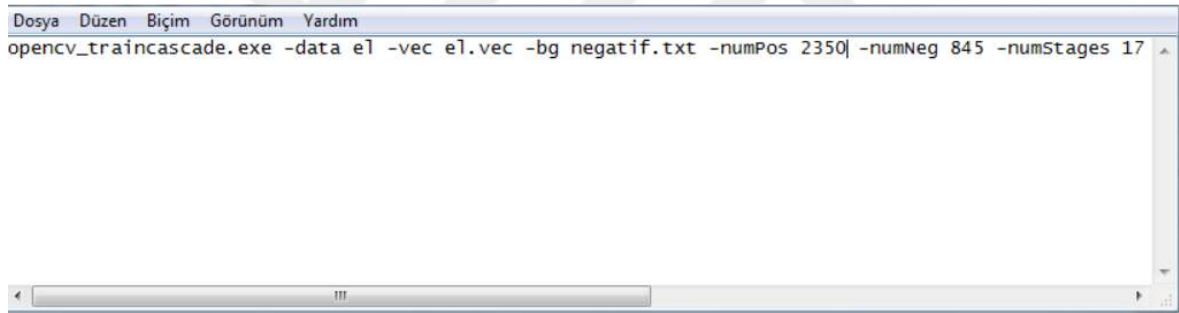


Şekil 3.12: Dosya Doğruluk Kontrolü

Gelen ekranda nesnenizin belirtilen koordinatlarda kırılmış ve yeniden boyutlandırılmış hali görülür. Burada koordinat değerleri yanlış bir örnek varsa bu sınıflandırıcınızın iyi çalışmamasına neden olacaktır.

6-Son olarak sınıflandırıcımızın eğitim aşaması vardır. Bu eğitim opencv kütüphanesinde bulunan iki .exe dosyası ile yapılabilmektedir. Bu dosyalar opencv_haartraining.exe ve opencv_traincascade.exe dosyalarıdır. opencv_haartraining.exe sadece haar yöntemini kullanan eski algoritmaları kullanmaktadır. opencv_traincascade.exe ise en güncel sınıflandırıcı eğitim dosyası olup hem haar yöntemi hem de local binary yönteminde eğitim yapabilmektedir. Bu .exe dosyası belirlenen aşama kadar pozitif resimleri negatif resimler üzerinde işleyerek belirlenen minhitrate ve maxfalsealarmrate değerlerine ulaştırmaya çalışır. Her bir aşamada pozitif resimler farklı çerçevelerle taranarak değerler oluşturulur. Belirlenen aşama değerine ulaşıldığında ise nesneyi bulmak için kullanılabilecek .xml dosyası oluşacaktır.

Ana dizinde yeni bir .bat dosyası oluşturulur ve düzenle kısmından opencv dizininde bulunan opencv_traincascade.exe dosyasını yoluyla beraber yapıştırılır ve şekil 3.13’ de gösterilen ifadedeki parametreler eklenir.



Şekil 3.13: Eğitim aşaması

” D:\opencv\build\x64\vc11\bin\opencv_traincascade.exe -data Cascade -vec el.vec -bg negatif.txt -numPos 2350 -numNeg 845 -numStages 17 -minHitRate 0.995 -maxFalseAlarmRate 0.25 -mem 1024 -w 24 -h 24 ” . Bu ifadede parametreleri oluşturduğumuz .bat dosyasına yapıştırıp çalıştırdığımızda belirli bir süre sonra .xml dosyasını oluşturur. Bu parametreler sınıflandırıcımızın performansının iyi olmasında çok etkilidir. Parametreler şu şekildedir:

- “D:\opencv\build\x64\vc11\bin\opencv_traincascade.exe” bu kısım opencv_traincascade.exe dosyasının bulunduğu dizini gösterir. Command Window bu komutla bu dosyayı çalıştırır ve yanda belirtilen değerleri de parametre olarak alır.

- "-data Cascade" bu kısımda sınıflandırıcının her bir aşamada(stage) elde ettiği xml dosyasının, parametrelerin tutulduğu xml dosyasının ve en son elde edilen xml dosyasının oluşturulduğu klasörün ismi belirtilir. En son cascade.xml isimli bir dosya oluşacaktır ve bu nesne bulmak ve takip etmede kullanılacaktır.
- "-vec el.vec" bu kısım pozitif resimlerle elde ettiğiniz vektör dosyasını tanımlar.
- "-bg negatif.txt" Bu kısımda 'bg' ingilizce background yani arka plan'ı gösterir. Pozitif resimlerin işleneceği arka plan resimlerinin dosya yerleriyle isimlerini kaydettiğimiz "negatif.txt" dosyasını parametre olarak veririz.
- "-numPos 2350" bu kısım, eğitimin aşamalarında kaç adet pozitif örnek kullanılacağını belirtir. Bu kısım .vec dosyasındaki toplam pozitif resim sayısı ile karıştırılmamalıdır. Eğer buraya vektör dosyasındaki pozitif resim sayısı girilirse hata oluşur; çünkü vec file, pozitif örneklerin negatif resimler sayısının kısmını arka planda işleyecektir ve minhitrate oranı da pozitif örnek sayısını sınırlandırır. Bu yüzden kullanılan pozitif örnek sayısı için bir formül geliştirilmiştir.
$$\text{numPos} \leq (\text{Vektör dosyasında toplam pozitif örnek sayısı} - \text{Negatif örnek sayısı}) / (1 + (\text{aşama sayısı} - 1) * (1 - \text{minhitrate}))$$
 formülüne göre en yüksek numPos oranı seçilmelidir. Örneğin vektör dosyasında 3400 örnekten oluşuyorsa ve Negatif örnek sayısı 845 ise, minhitrate sayısını 0.995 ve aşama sayısını 17 alınırsa
$$\text{numPos} < (3400 - 845) / (1 + 16 * 0,005)$$
 formülüne göre numPos < 2365 çıkacaktır. Bu değerden küçük olarak 2350 sayısı numPos olarak seçilebilir.
- "-numNeg 845" bu kısım kaç tane negatif örnek resim kullanılacağını ifade etmek için kullanılır. "negatif.txt" dosyasındaki negatif resim sayısı kullanılabilir.
- "-numStages 17" bu kısımda sınıflandırıcının kaç aşamada sonuca ulaşacağı değeri belirtilir.
- "-minHitRate 0.995" her bir aşamada kullanılan minimum isabet oranını verir. Genel olarak 0.995 değeri verilir.
- "-maxFalseAlarmRate 0.25" nesneleri tanırken kabul edilebilir maksimum hata oranını belirtir. Eğer bu oran yüksek olursa sınıflandırıcı, az benzettiği nesneleri de aradığımız nesne olarak sunacaktır. Çok küçük olursa da istediğimiz nesnenin küçük açılarda farklılıklarında bile bulamayacaktır. Bu oranı ayarlamak kullanım alanına göre değişiklik gösterecektir.
- "-mem 1024" sınıflandırıcının eğitimi için ayrılan belleği belirtir.

- "-w 24 -h 24" vektör dosyasının boyutunu verir. Bu boyut vektör dosyasını oluştururken kullandığınız boyutla aynı olmalıdır.

Parametreleri ayarladıktan sonra sınıflandırıcı eğitimi için oluşturduğumuz .bat dosyasını çalıştırdığımızda sonuç şekil 3.14'de ki gibi olacaktır. Bu eğitimde karşılaşılabileceğimiz hataların büyük kısmı numPos hesabından olacaktır. Eğer hata alınıyorsa numPos değerini büyük bir oranda azaltıp deneme yapılmalıdır. Eğer çok örnek kullanılırsa ve düşük bir maxfalsealarmrate oranı seçersek sınıflandırıcının eğitimi çok uzun sürecektir.

Bu şekilde kendi sınıflandırıcınızı oluşturabilir, istediğimiz nesneyi seçebiliriz; ancak reel time çalışmalarda nesne bulma kısmı daha çok kaynak gerektirmektedir [58].

Bu çalışmada yüz tespiti için Intel tarafından geliştirilmiş olan haarcascade_frontalface_default.xml dosyası kullanılmıştır. Bu dosya görüntü işleme kütüphaneleri için açık kaynak kodlu bir dosyadır. Bu .xml dosyası yaklaşık 10000 adet eğitim verisinden oluşmaktadır. Uygulamamızın ana dizinine bu dosyayı ekleyerek yüz tespiti için kullanılabilir duruma getirilmiştir.

Şekil 3.14: Eğitici .bat dosyasının çalıştırılması

4. UYGULAMA

4.1 Uygulama İçin Kullanılan Yazılım ve Kütüphaneler

Bu çalışmada gerçekleştirilen “Gerçek ve yarı gerçek zamanlı Yüz tespit” uygulaması Microsoft Visual Studio 2015 ortamında C# dili yazılmıştır.

Proje kapmasında bilgisayarlı görüntü işleme işlemlerini hızlandırmak amacıyla “AForge.NET” isimli yazılım kütüphanesi kullanılmıştır. “EmguCV” yazılım kütüphanesi, bilgisayarlı görüntü işleme çalışmalarında sıklıkla kullanılan OpenCV yazılım kütüphanesinin “.NET çerçevesi” için hazırlanmış çevreleyicisidir (wrapper). “AForge.NET” ile bilgisayarlı görüntü işleme işlemlerinin yapılmasına rağmen kameradan görüntü almak amacıyla fazladan “thread” kullanması sürekli bilgilerin güncellendiği arayüzde takılmalara ve programın yanıt vermemesine sebep olmaktaydı. Özellikle bu sorunu ortadan kaldırmak için “EmguCV” kullanılmış ve “thread” mantığı terk edilerek, “zamanlayıcı (timer)” mantığı benimsenmiştir. “EmguCV” kullanımının bir diğer faydası, işletim sistemi bağımlılığını ortadan kaldırmasıdır [59].

4.1.1 AForge.NET

"AForge.NET" yazılım çerçevesi, bilgisayarlı görüntü işleme ve yapay zekâ geliştiricileri, araştırmacıları için tasarlanmıştır. Yazılım çerçevesinin sahip olduğu kütüphaneler ve özellikleri:

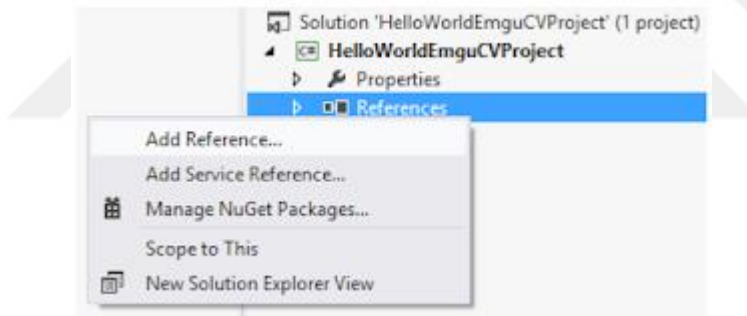
- AForge.Video: Video işleme kütüphanesi
- AForge.MachineLearning: Makine öğrenme kütüphanesi
- AForge.Imaging: Standart görüntü işleme ve filtreleme kütüphanesi
- AForge.Vision: Bilgisayarlı görüntü işleme kütüphanesi
- AForge.Robotics: Birkaç robot kitlerinin desteklenmesi için hazırlanan kütüphanesi
- AForge.Neuro: Yapay sinir ağları hesaplama kütüphanesi
- AForge.Genetic: Evrimsel programlama kütüphanesi

Projede videonun picturebox içerisinde gösterilmesinde Aforge Framework’nün AVIReader ve AForge.Video.VFW kütüphaneleri ve yüklenen görüntü üzerinde yüzün tespitinde EmguCV Framework’ü kullanılmıştır.

4.1.2 EmguCV

"EmguCV" işletim sistemleri arası taşınabilir, "OpenCV"nin ".NET" çerçevesi için hazırlanmış bir çevreleyicidir (wrapper). Bu sayede "OpenCV"nin fonksiyonlarını ".NET" destekli C#, VB, VC++, IronPython gibi dillerle de çalışması sağlanmıştır. Çevreleyici "Mono" çerçevesi ile "Linux" çekirdeğine sahip ya da "Mac OS X" işletim sistemlerinde de çalışabilir. "EmguCV"nin sahip olduğu temel katman "OpenCV"nin tüm yapılara destek vermektedir. İkincil katman sayesinde, var olan özelliklerin yanına ".NET" çerçevesinin getirdiği ek özelliklerde eklenmiştir. Açık kaynak kodlu ücretsiz bir yazılımdır.

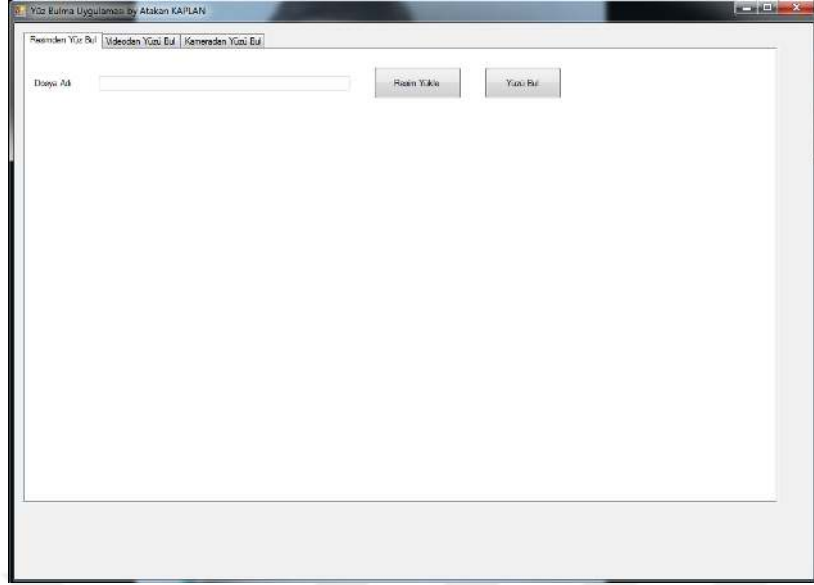
Uygulamamızda EmguCV dll'lerini indirmek için http://www.emgu.com/wiki/index.php/Main_Page adresinden dll dosyaları indirildikten sonra şekil 4.1'de ki gibi Visual Studio'da projemize referans olarak eklememiz gerekir.



Şekil 4.1 : dll dosyalarının eklenmesi

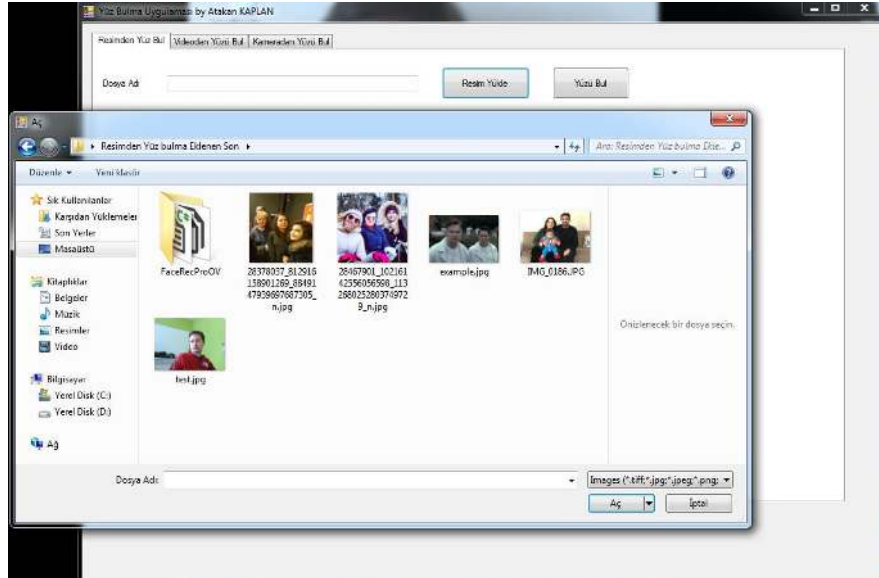
4.2 Geliştirilen Sistem

Uygulama 3 bölümden oluşmaktadır(şekil 4.2):



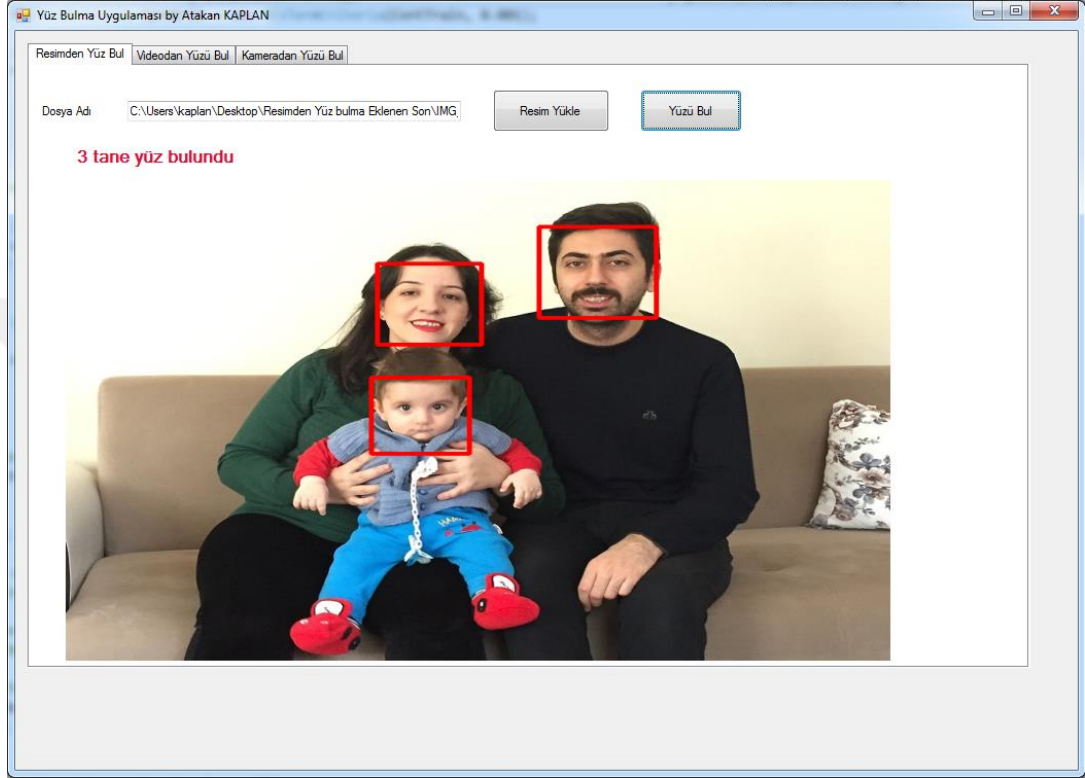
Şekil 4.2: Program arayüzü

1-Resimden Yüz Bul: Bu kısımda kullanılan resim formatları “.tiff;*.jpg;*.jpeg;*.png;*.gif;*.bmp” şeklindedir. Resim yükle butonuna basıldığında yüz tespiti yapmak istediğimiz resmi seçeriz(şekil 4.3).



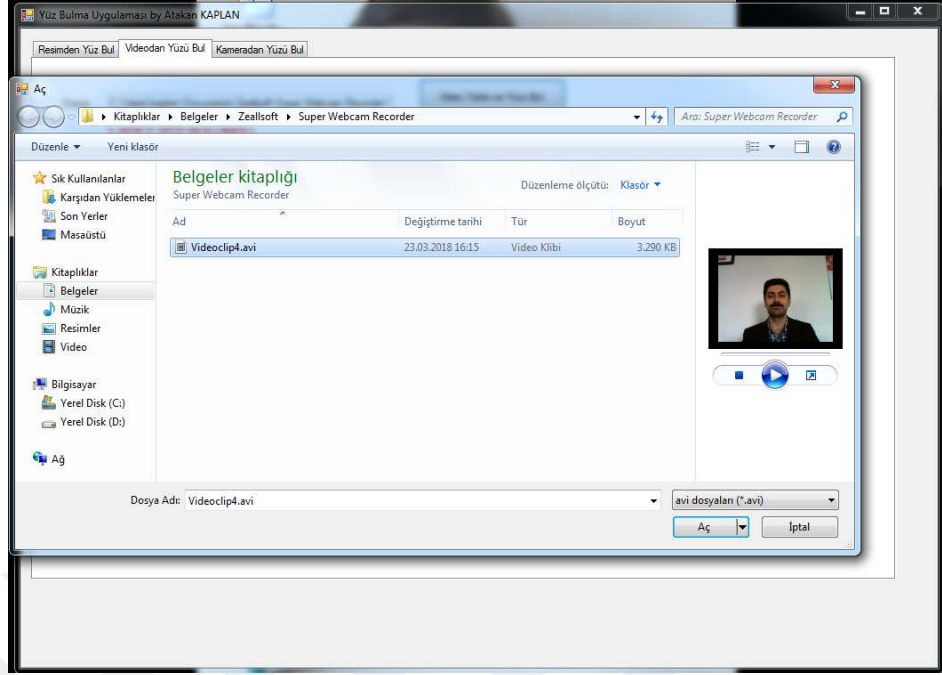
Şekil 4.3: Resim yükleme

Resmi yükleyip Yüzü Bul butonuna tıkladığımızda yüklenmiş olan resimdeki yüzleri bulup bize şekil 4.4’de ki gibi kaç adet yüz bulunduğunu ve bulunduğu yüzleri kırmızı dikdörtgen nesnesi içinde göstermektedir.



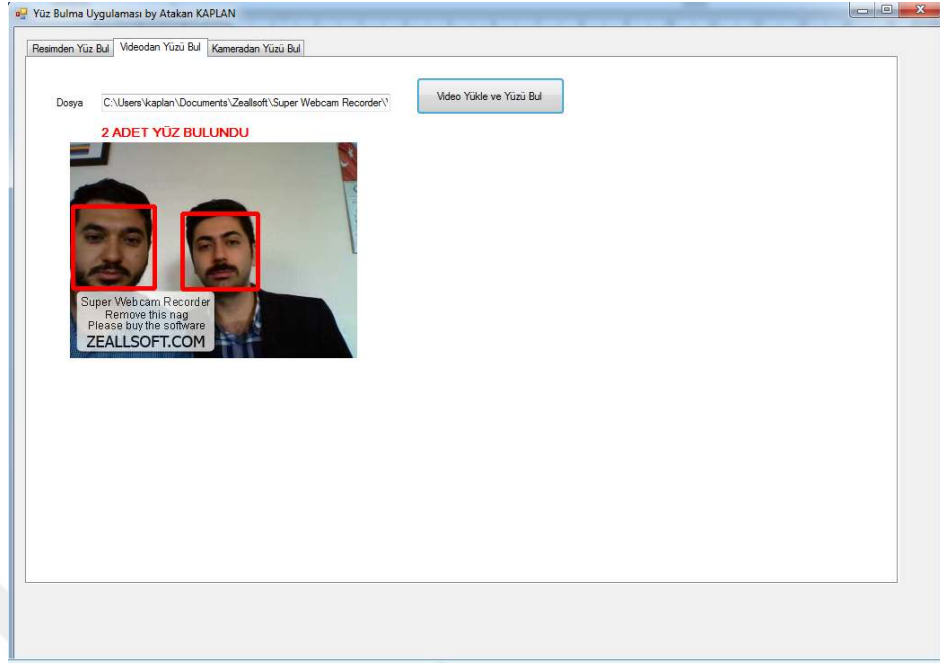
Şekil 4.4 : Yüzlerin bulunması

2-Videodan Yüz Bul: Bu kısmında video kayıtları AVI formatında saklanarak işlenmiştir. Tespit sonuçları da yine aynı şekilde AVI formatındadır. Uygulamada Video Yükle ve Yüzü Bul butonuna basıldığında şekil 4.5’de ki gibi video yolunu gösterilmesi istenecektir.

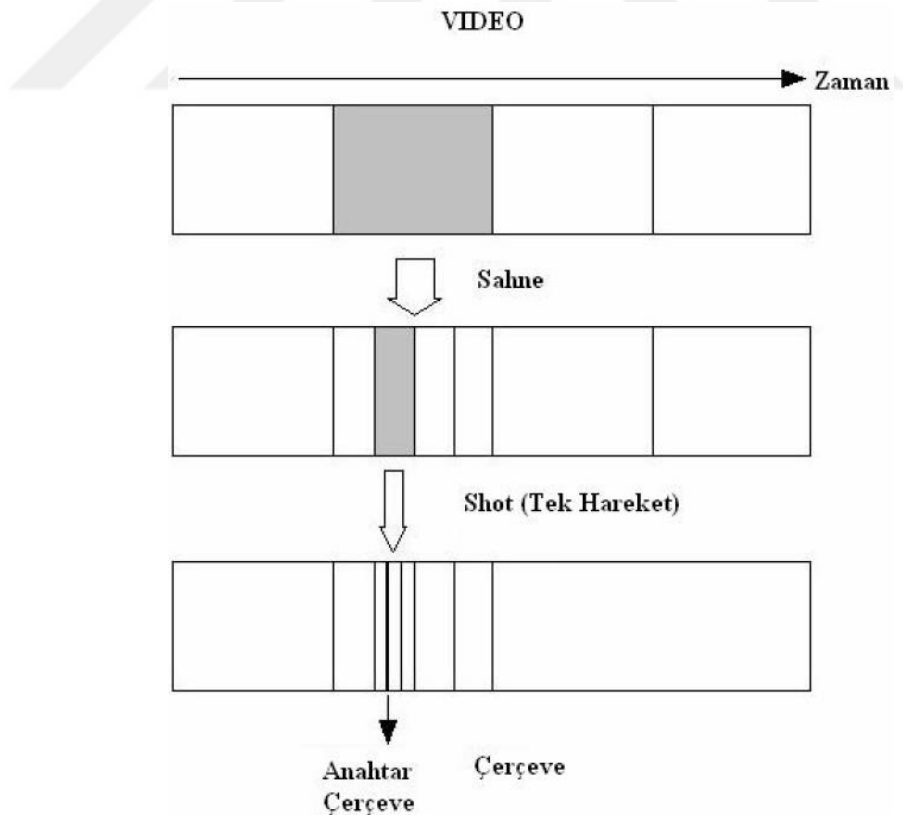


Şekil 4.5: Video Yükleme

Butona tıkladıktan sonra şekil 4.6’da ki gibi picturebox üzerinde video oynatılmaya başlanacak ve varsa birden fazla yüz bulunacaktır. Tespit edilen yüzler kırmızı dikkörtgen içinde algılanacak ve kaç adet olduğu gösterilecektir. Şekil 4.7’de görüldüğü gibi video verisini oluşturan resimlerden her birine çerçeve(frame) denilmektedir. Bir tek kameradan kaydedilmiş, ardışık sıralı çerçevelerden oluşan ve video akımını oluşturan bloklara shot (videodaki tek hareket) denilmektedir. Shot’ın içeriğini en iyi temsil eden çerçeveye de anahtar çerçeve denilmektedir. Shot’lar dillerdeki harflere benzerler; tek baslarına anlam ifade etmezler, yan yana geldiklerinde anlam kazanırlar. Shot’ların bir araya gelmesi ile oluşan anlamlı video bölümlerine sahne denilmektedir. Shot’lar fiziksel sınırları, sahneler ise semantik sınırları göstermektedir [60].

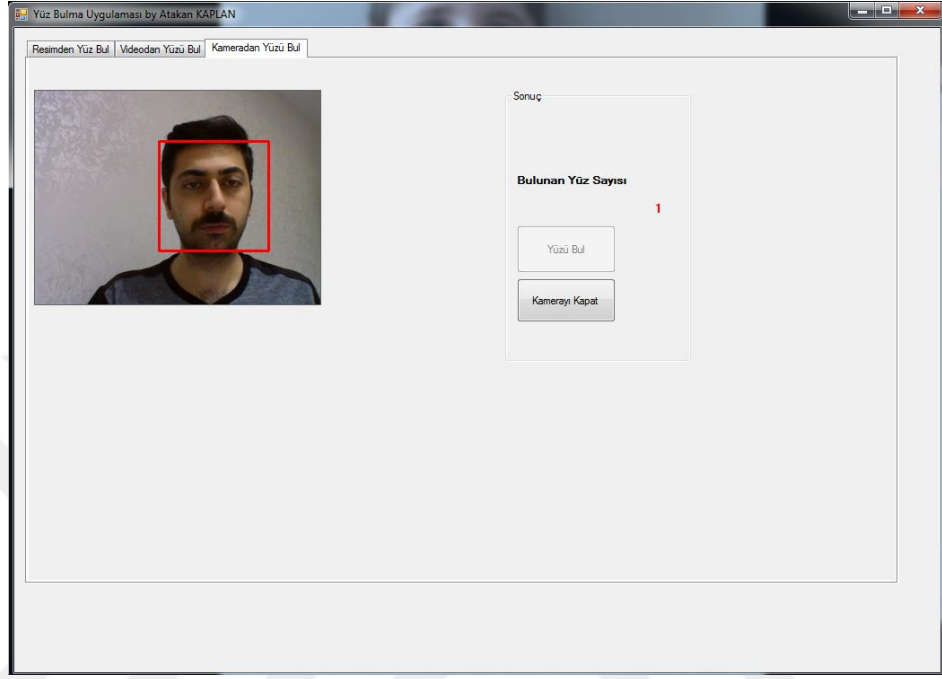


Şekil 4.6: Video dosyasında yüz tespiti



Şekil 4.7 : Video akımının yapısı

3-Kameradan Yüz Bul kısmında notebook üzerinden bulunan dahili kameradan görüntü elde edilip yüz tespit işlemi şekil 4.8’de ki gibi gerçekleştirilmiştir.



Şekil 4.8 : Kameradan yüz tespit

Uygulamanın her 3 bölümünde de gerçekleşme aşamasında yüz tespit edilirken resimler şekil 4.9’da ki gibi gri-seviye dönüşümü yapılarak 20x20 ebatlarında yeniden boyutlandırılmıştır.



Şekil 4.9 : gri-seviye dönüşümü yapılmış resim

5. SONUÇ

Bu çalışmada, görüntü işleme teknikleri kullanılarak bir görüntünün yüz içerip içermediğinin tespiti ve resim üzerinde yüz yerlerini saptama işlemleri gerçekleştirilmiştir. İşlemler gerçekleştirilirken öncelikle görüntüler normalize edilmiştir(gri-seviye dönüşüm). Çalışma kapsamında video tabanlı ve gerçek zamanlı bir yüz tespiti için ihtiyaç duyulan metodolojiler ve yaklaşımlar tasarlanmış ve problemin çözümü için yüksek standartlarda bir uygulama geliştirilmiştir. Bu konularda yapılan pek çok bilimsel makale ve tez çalışmasında yüz bulma konuları birbirlerinden bağımsız olarak incelenmiştir. Yüz tespitinde poz farklılıklarından kaynaklanan dezavantajlardan kurtulmak için kullanmış olduğumuz .xml dosyasını daha fazla örnekle eğitimi gerçekleştirilebilir. Ayrıca farklı teknikler kullanılarak bu dezavantajların önüne geçilebilir. Bu tez çalışmasında sonuç olarak yüz bulma sistemi gerçek zamanlı bir uygulamanın ihtiyaçlarını ve hedeflerini karşılayacak şekilde geliştirilmiştir. Bu yöntemler geliştirilirken uygulamanın gerçek zamanlı bir uygulama olarak yüksek performans ve hızla çalışması gerektiği de hiçbir zaman gözden kaçırılmamıştır.

KAYNAKLAR

- [1] Elman, J.L., E.A. Bates, M.H. Johnson, A. Karmiloff-Smith, D. Parisi, K. Plunkett, Rethinking innateness: A connectionist perspective on development, *Cambridge, MA: MIT Press*, 1996.
- [2] Chellappa, R., C.L. Wilson, S. Sirohey, "Human and machine recognition of faces: a survey", *Proceedings of the IEEE*, vol.83, no.5, 1995.
- [3] Turk, M., A. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol.3, no.1, pp.71-86, 1991.
- [4] Wiskott, L., J.-M. Fellous, N. Krüger, C. von der Malsburg, "Face recognition by elastic bunch graph matching", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.19, no.7, pp.775-779, 1997
- [5] Ramasubramanian, D., Y.V. Venkatesh, "Encoding and recognition of faces based on the human visual model and DCT", *Pattern Recognition*, vol.34, pp.2447-2458, 2001.
- [6] Tanaka, J. W., Farah M.J., "Parts and wholes in face recognition," *Quarterly Journal of Experimental Psychology: Human Experimental Psychology*, vol.46A, 225-245, 1993.
- [7] Fodor, J.A., The Modularity of Mind, *MIT Press, Cambridge, MA*, 1983.
- [8] Karmiloff-Smith, A., Beyond modularity: A developmental perspective on cognitive science, *Cambridge MA: MIT Press*, 1992.
- [9] Süleyman, F., 2012. Siber Güvenlikte Biyometrik Sistemler ve Yüz Tanıma, *Yüksek Lisans Tezi*, Gazi Üniversitesi Bilişim Enstitüsü, Ankara
- [10] Eleyan, G., 2010. Görüntü Çerçevelerinde Yüz Algılama ve Veritabanı ile Eşleme Yapılması, *Yüksek Lisans Tezi*, Ankara Üniversitesi Elektronik Mühendisliği Anabilim Dalı, Ankara
- [11] Kurt, B., Bilgisayar ile Psikolojik Durum Değerlendirmesi, Yüksek Lisans Tezi, KTÜ Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, 2007
- [12] <http://mehmetsalihdeveci.net/2010/12/31/yuzresimlerinde> Özellik Noktalarının Bulunması.2012
- [13] Özmen , G., Kandemir , R., 2012. Haar Dalgacıkları ve Kübik Bezier Eğrileri İle Yüz İfadesi Tespiti, ELECO '2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu, Bursa, 29 Kasım - 01 Aralık
- [14] M. Pantic, and L. J. M. Rothkrantz, *IEEE Trans. on PAMI*, 22, 1424(2000).

- [15] **P. Viola and M. Jones.**, Robust real-time object detection, Technical Report CRL 20001/01, Cambridge Research Laboratory, Cambridge Inc., 2001.
- [16] **M. Yang, D. Kriegman, and N. Ahuja**, IEEE Trans. on Pattern Analysis and Machine Intell., 24, 34(2002).
- [17] **Şakar, M.**, 2011. İnsan Yüz İmgelerinden Yüz İfadesi Tanıma, *Yüksek Lisans Tezi*, Fırat Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı, Elazığ
- [18] **Yang, G. and Huang, T. S.**, 1994. Human Face Detection in Complex Background, Pattern Recognition, vol. 27, no. 1, pp. 53-63.
- [19] **Kotropoulos, C. and Pitas, I.**, 1997. Rule-Based Face Detection in Frontal Views, Proc. Int'l Conf. Acoustics, Speech and Signal Processing, vol. 4, pp. 2537-2540.
- [20] **Sirohey, S.A.**, 1993. Human Face Segmentation and Identification, Technical Report CSTR- 3176, University of Maryland.
- [21] **Leung, T.K., Burl, M.C. and Perona, P.**, 1995. Finding Faces in Cluttered Scenes Using Random Labeled Graph Matching, Proc. Fifth IEEE Int'l Conf. Computer Vision, pp. 637-644.
- [22] **Burl, M.C., Leung, T.K. and Perona, P.**, 1995. Face Localization via Shape Statistics, Proc. First Int'l Workshop Automatic Face and Gesture Recognition, pp. 154-159.
- [23] **Leung, T.K., Burl, M.C. and Perona, P.**, 1998. Probabilistic Affine Invariants for Recognition, Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 678-684.
- [24] **Satoh, S., Nakamura, Y. and Kanade, T.**, 1999. Name-It: Naming and Detecting Faces in News Videos, IEEE Multimedia, vol. 6, no. 1, pp. 22-35.
- [25] **Crowley, J.L. and Berard, F.**, 1997. Multi-Modal Tracking of Faces for Video Communications," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 640-645.
- [26] **Sobottka, J. and Pitas, I.**, 1996. Segmentation and Tracking of Faces in Color Images, Proc. Second Int'l Conf. Automatic Face and Gesture Recognition, pp. 236-241.
- [27] **Chai, D. and Ngan, K.N.**, 1998. Locating Facial Region of a Head-and-Shoulders Color Image. 3rd International Conference on Face & Gesture Recognition, pp. 124-129, Japan.
- [28] **Saber, E. and Tekalp, A.M.**, 1998. Frontal-View Face Detection and Facial Feature Extraction Using Color, Shape and Symmetry Based Cost Functions, Pattern Recognition Letters, vol. 17, no. 8, pp. 669-680.

- [29] Tsukamoto, C., Lee, W. and Tsuji, S., 1994. Detection and Pose Estimation of Human Face with Synthesized Image Models, Proc. Int'l Conf. Pattern Recognition, pp. 754-757
- [30] Özmen, G., 2012. Kübik Bezier Eğrileri ile Yüz İfadesi Tanıma, *Yüksek Lisans Tezi*, Trakya Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı, Edirne
- [31] Sung, K. ve Poggio, T., (1998), "Example-based Learning for View-based Face Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, (20):39-51.
- [32] Eser, S., Yapay Sinir Ağları İle Yüz Sezimi ve Takibi, Yüksek Lisans Tezi, YTU, 2006
- [33] Goldstein, A.O., Harmon, D., and Lesk, B., 1971. Identification of Human Faces. Proceedings of the IEEE, Vol. 59, no. 5.
- [34] Sirovich, L. and Kirby, M., 1987. Low-dimensional procedure for the characterization of human faces. Optical Society of America. Vol. 4, pp: 519- 525.
- [35] Turk, A. and Pentland, P., 1991. Face Recognition Using Eigenfaces. Media Lab., MIT, Cambridge, MA, USA. IEEE. pp: 586 – 591.
- [36] Vetter T., and Poggio, T., 1997. Linear Object Classes and Image Synthesis From a Single Example Image.
- [37] Sherrah, J., Gong, S. and Ong, E.J., 2000. Face distribution in similarity space under varying head pose. Image and vision computing. pp: 807-819.
- [38] Bae, H., and Kim, S., 2005. Real-time face detection and recognition using hybrid information extracted from face space and facial features. Image and Vision Computing. pp: 1181–1191.
- [39] Ahonen T., Jadod, A., and Pietika, M., 2006. Face Description with Local Binary Patterns: Application to Face Recognition. IEEE transactions on pattern analysis an machine intelligence, Vol. 28.
- [40] Yang, J., Ling, X., Zhu, Y. and Zheng, Z., 2008. A face detection and recognition system in color image series. Mathematics and Computers in Simulation 77 . pp: 531–539.
- [41] Singh, R., Vatsa, M. and Noore, A., 2009. Face recognition with disguise and single gallery images. Image and Vision Computing. pp: 245–257.
- [42] Wang, J., Yang, W. and Yang, J., 2009. Face recognition using fuzzy maximum scatter discriminant analysis. Springer-Verlag. Neural Comput & Applic. pp: 1- 4.
- [43] Kaminski, J., Knaan, D. and Shavit, A., 2009. Single image face orientation and gaze detection. Machine Vision and Applications. Vol. 21, pp: 85–98.

- [44] Echardt, M., Fasel, L., and Movellan, J., 2009. Towards practical facial Feature detection. *International journal of pattern recognition And artificial intelligence*. Vol. 23, no. 3. pp: 379–400.
- [45] Kim, J., Kim, K.H. and Song, M.K., 2009. Real Time Face Detection and Recognition Using Rectangular Feature Based Classifier and Modified Matching Algorithm. *IEEE*. pp: 171-175.
- [46] Shi, K., Pang, S. and Yu1, F., 2012. A Real-time Face Detection And Recognition System. *Shenzhen Inst. of Adv. Technol., Shenzhen, China*. pp: 3074 – 3077
- [47] Li, X., Fei, S. and Zhang, T., 2009. Median MSD-based method for face recognition. *Neurocomputing*. pp: 3930–3934.
- [48] Mishra, A., Swain, M. and Das, B., 2012. An approach to face recognition of 2-d images using eigen faces and PCA. *Signal & Image Processing : An Internationa Journal (SIPIJ)* Vol.3, No.2
- [49] Pujol, F. and Garcia, J., 2012. Computing the Principal Local Binary Patterns for face recognition using data mining tools. *Expert Systems with Applications*. pp: 7165-7172.
- [50] Shokouh, S.G., 2013. Gerçek Zamanlı Sayısal Görüntü İşleme ve Örüntü Tanıma Tekniklerinin Araştırılması ve Uygulanması, *Yüksek Lisans Tezi*, Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı, Ankara
- [51] P. Viola, M. Jones, "Robust real-time face detection," in *Internation Journal of Computer Vision*, 2004.
- [52] P. Viola, M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *Computer Vision and Pattern Recognition*, 2001.
- [53] Mihçioğlu, E.M., 2016. Gerçek Zamanlı Uygulamalar İçin Yaya Tespit Sistemi, *Yüksek Lisans Tezi*, Hacettepe Üniversitesi Elektrik Ve Elektronik Mühendisliği Anabilim Dalı, Ankara
- [54] Porwik, P., Lisowska, A., 2004, The Haar–Wavelet Transform in Digital Image Processing: Its Status and Achievements, *Machine graphics & vision*, 13 (1-2), 79-98.
- [55] Gökmen, M., Kurt, B., Kahraman, F., Çapar, A., 2007, Çok Amaçlı Gürbüz Yüz Tanıma, *İstanbul Teknik Üniversitesi*, 104E121.
- [56] Özyurt, O., 2012. Hareketsiz Hastalar İçin Gerçek Zamanlı Göz Kırpma İletişim Sistemi Tasarımı ve Uygulanması, *Yüksek Lisans Tezi*, Selçuk Üniversitesi Elektronik ve Bilgisayar Sistemleri Anabilim Dalı, Konya

- [57] **Ayvaz , U., Gürüler , H.,** 2017. Bilgisayar Kullanıcılarına Yönelik Duygusal İfade Tespiti, bilişim teknolojileri dergisi, cilt: 10, sayı: 2
- [58] <http://talhakoc.net/> OpenCV Haar Cascade sınıflandırıcısı. 2014.
- [59] **Keskin, O.,** 2009. Kameralı Görüntü Alanı İçinde Hareket Kontrolü, *Yüksek Lisans Tezi*, Dokuz Eylül Üniversitesi Mekatronik Mühendisliği Anabilim Dalı, İzmir
- [60] **Hamamcı, S.,** 2007. Görüntü Özelliklerinin Değişimine Göre Haber Videolarının Özetlenmesi, *Yüksek Lisans Tezi*, Yıldız Teknik Üniversitesi FBE Bilgisayar Mühendisliği Anabilim Dalı, İstanbul



ÖZGEÇMİŞ

Atakan KAPLAN 15.11.1988 yılında Elazığ’da doğdu. İlk, orta ve lise öğrenimini Elazığ’da tamamladı. 2006 – 2010 yılları arasında Fırat Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde okudu. 2010 yılında aynı bölümden bilgisayar mühendisi olarak mezun oldu. 2010 – 2011 yılları arasında Fırat Üniversitesinde burslu öğrenci olarak çalıştıktan sonra 2012 yılında Bingöl İl Sağlık Müdürlüğünde Bilgisayar Mühendisi olarak göreve başladı. 2015 yılında bulunduğu kurumdan ayrılarak Elazığ İl Sağlık Müdürlüğünde çalışmaya başladı. 2010 yılında Fırat Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği (yazılım) bölümünde yüksek lisans eğitimine başladı. Halen Elazığ İl Sağlık Müdürlüğünde çalışmakta olup aynı zamanda Fırat Üniversitesi’nde yüksek lisans öğrencisi olarak devam etmektedir.